

2

NAVAL POSTGRADUATE SCHOOL

Monterey, California

AD-A261 827



THESIS

DTIC
ELECTE
MAR 19 1993
S E D

THE DESIGN AND IMPLEMENTATION OF
A REUSABLE COMPONENT LIBRARY AND
A RETRIEVAL/INTEGRATION SYSTEM

by

Dogan Ozdemir

December 1992

Thesis Advisors:

Yuh-jeng Lee
Luqi

Approved for public release; distribution is unlimited.

93-05820



REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6a. NAME OF PERFORMING ORGANIZATION Computer Science Dept. Naval Postgraduate School	6b. OFFICE SYMBOL (if applicable) CS	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Ada Joint Program Office	8b. OFFICE SYMBOL (if applicable)	10. SOURCE OF FUNDING NUMBERS	
8c. ADDRESS (City, State, and ZIP Code) The Pentagon, Room 3E114 Washington DC, 20301-3081		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) THE DESIGN AND IMPLEMENTATION OF A REUSABLE COMPONENT LIBRARY AND A RETRIEVAL/INTEGRATION SYSTEM (U)			
12. PERSONAL AUTHOR(S) Dogan Ozdemir			
13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED FROM 07/90 TO 12/92	14. DATE OF REPORT (Year, Month, Day) December 1992	15. PAGE COUNT 345
16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
		Reusability, Reusable Software Components, Component Retrieval and Integration, Computer Aided Prototyping, Graphical User Interface.	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) Rapid prototyping with automated retrieval of reusable software component is a software development method to construct software systems expeditiously. This thesis describes a tool to enhance the practice of software reuse within the Computer Aided Prototyping System (CAPS). A software base interface provides prototype designers with the means to retrieve components and integrate them into new applications. Reusable components are retrieved from the software base using a formal specification as the search key or through a browser. The specification language used is the Prototype System Description Language (PSDL). The software base stores the reusable components in an object oriented database management system (ONTOS) with an appropriate PSDL specification. Following a query conducted by the PSDL specification, chosen retrieved components are transformed and integrated to the system under development. All software base procedures, including the storage, retrieval, and integration of the components, are conducted through a graphical user interface which is designed to demonstrate and manipulate available software base operations.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Yuh-jeng Lee		22b. TELEPHONE (Include Area Code) (408) 646-2361	22c. OFFICE SYMBOL CS/LE

Approved for public release; distribution is unlimited

**THE DESIGN AND IMPLEMENTATION OF
A REUSABLE COMPONENT LIBRARY AND
A RETRIEVAL/INTEGRATION SYSTEM**

by

Dogan Ozdemir
Lieutenant JG, Turkish Navy
B.S., Turkish Naval Academy, 1986

Submitted in partial fulfillment of the
requirements for the degree of

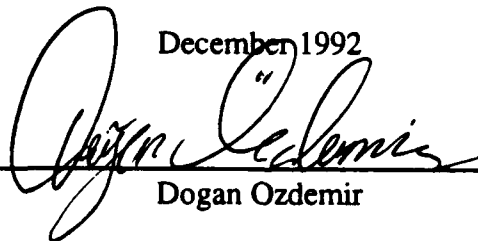
MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL

December 1992

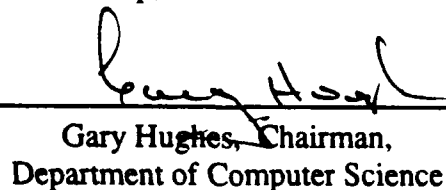
Author:


Dogan Ozdemir

Approved By:


Yuh-jeng Lee, Thesis Advisor


Luigi, Second Reader


Gary Hughes, Chairman,
Department of Computer Science

ABSTRACT

Rapid prototyping with automated retrieval of reusable software components is a software development method to construct software systems expeditiously. This thesis describes a tool to enhance the practice of software reuse within the Computer Aided Prototyping System (CAPS).

A software base interface provides prototype designers with the means to retrieve components and integrate them into new applications. Reusable components are retrieved from the software base using a formal specification as the search key or through a browser. The specification language used is the Prototype System Description Language (PSDL). The software base stores the reusable components in an object oriented database management system (ONTOS) with an appropriate PSDL specification. Following a query conducted by the PSDL specification, chosen retrieved components are transformed and integrated to the system under development.

All software base procedures, including the storage, retrieval, and integration of the components, are conducted through a graphical user interface which is designed to demonstrate and manipulate available software base operations.

DTIC QUALITY INSPECTED 1

Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification:	
By	
Distribution /	
Availability Codes	
Dist	Avail and / or Special
A-1	

TABLE OF CONTENTS

I. INTRODUCTION	1
A. SOFTWARE REUSE	1
B. RAPID PROTOTYPING	4
C. THE COMPUTER AIDED PROTOTYPING SYSTEM (CAPS)	6
D. THE PROTOTYPE SYSTEM DESCRIPTION LANGUAGE (PSDL)	7
E. OBJECTIVES	9
F. ORGANIZATION OF THE THESIS	9
II. BACKGROUND AND PREVIOUS RESEARCH	10
A. INFORMATION RETRIEVAL	10
B. METHODS FOR RETRIEVING REUSABLE COMPONENTS	11
C. CURRENT RETRIEVAL SYSTEMS AND TOOLS	14
III. CAPS SOFTWARE BASE	18
A. COMPONENT STORAGE	19
B. RETRIEVING THE COMPONENTS	22
IV. INTEGRATION OF REUSABLE COMPONENTS	24
A. INTEGRATING THE COMPONENTS INTO CAPS	24
B. METHODOLOGY FOR TRANSFORMATION	25
C. STRUCTURE OF THE INTEGRATION TOOL	26
V. GRAPHICAL USER INTERFACE FOR SOFTWARE BASE	37
A. DESIGN PRINCIPLES	37
B. IMPLEMENTATION CONSIDERATIONS	38
C. SOFTWARE BASE GUI ARCHITECTURE	40

D. CONCLUSIONS	42
VI. CONCLUSIONS AND FUTURE RESEARCH	44
A. ACCOMPLISHMENTS	44
B. FUTURE WORK	44
LIST OF REFERENCES	47
APPENDIX A - INTEGRATION EXAMPLES	50
A. INTEGRATING A COMPONENT RETRIEVED BY NAMED LOOKUP	50
B. INTEGRATING A COMPONENT RETRIEVED BY PSDL QUERY	52
APPENDIX B - SOFTWARE BASE GUI USER'S MANUAL	55
A. LIBRARY FUNCTIONS	56
B. MAIN MENU	58
C. VIEW COMPONENT	62
D. SELECT COMPONENT	63
APPENDIX C - SOURCE CODE FOR INTEGRATION TOOL	66
A. psdl_lex.l	66
B. psdl.y	69
C. query_lex.l	80
D. query.y	83
E. component_aux_pkg	94
F. component_compose_pkg	100
G. component_parse.a	110

H. query_aux_pkg	111
I. query_compose_pkg	117
J. query_parse.a	127
APPENDIX D - SOURCE CODE FOR GUI	129
A. softbase.a	129
B. global package	134
C. panel_library package	143
D. panel_libadd package	152
E. panel_lbdelete package	163
F. panel_ldelwarn package	173
G. panel_lbselect package	183
H. panel_mainmenu package	193
I. panel_addfile package	206
J. panel_keyword package	225
K. panel_query package	236
L. panel_compsel package	248
M. panel_viewpsdl package	259
N. panel_viewspec package	268
O. panel_viewbody package	277
P. panel_select package	285
Q. panel_savecomp package	298

R. panel_cdelwarn package	310
S. panel_integrate package	319
T. softbase_support package	330
U. softbase_creat_init.a	335
INITIAL DISTRIBUTION LIST	336

I. INTRODUCTION

A. SOFTWARE REUSE

1. What Is Software Reuse

Software reuse is using the elements of an existing system in the construction of another similar system.

In general, software reuse includes all the products of software development cycle. Software documentation, requirements, analyses, designs, source codes etc. can all be reused in this broad view.

Reuse of software is an important issue in software engineering because developers benefit in both productivity and accuracy by reusing software entities from all phases of system development.

2. Why Reuse

Much has been written about the software crisis in recent years. It was stated that software is too costly, insufficient quality and its development is nearly impossible to manage [Cox90]. Software reuse may reduce the pressure of the software crisis.

A great percentage of a typical program is composed of potentially reusable code. It is desirable to make use of existing code wherever possible since this can have a significant impact on both the cost and duration of software development.

Automating the reusability of code will help to eliminate the, now infamous, software backlogs [Bass87]. Furthermore, design analysis can focus on meeting and matching user requirements rather than verifying internal inconsistencies, especially in large projects.

It was reported that 90% of the cost of maintenance can be reduced by software reuse, another report described that the cost of software maintenance is usually about 75% of the total cost of the whole life-cycle of software. Other studies show that 40% of a design and 75% of the code on a given project is reused [KL91].

As can be seen, reuse can significantly reduce the cost of development and maintenance. Other than cost, software reuse may help improve the quality of the software. Frequent reuse of a software component may lead to frequent evaluation and revision. Therefore, high quality components may be constructed through the reuse process.

Reusing the software also saves time in all stages of the life cycle, makes learning new systems easier, improves readability and robustness, and leads to more modular, rational designs.

3. How To Reuse

While software reuse appears to be very beneficial it has not been widely used. It is only recently that emphasis is given to develop software systems that utilizes reusability.

Software reuse brings many research questions into the open. The arguments focus on the question that what the candidates are for software reuse, how reusable software components should be stored, how we can locate reusable software components, and how we can incorporate reusable software components into our own software system [YT88].

The candidates for software reuse brings the classification of software reuse. One example classification schema is reuse-in-the-small and reuse-in-the-large [KL91]. Reuse-in-the-small is concerned with the reuse of (small) pieces of source code such as classes, subroutines, (Ada) packages and so on. Reuse-in-the-large is concerned both with the reuse of large-grain components such as subsystems, and the reuse of elements beyond source code such as design structure, design decision, domain knowledge, analysis information and so forth.

This thesis will address only the source code reuse schemes.

Many kinds of mechanisms have been thought for the storage, retrieval and incorporation of components into the software. One of the earlier methods is cut and paste mechanism from component libraries [FH87]. The reusable components were stored in reference libraries and retrieved by various browsing mechanisms. Once the component is

found, the user should copy and paste the component information into his/her program, sometimes through the templates that was provided for this purpose.

An alternative view is that reuse is part of a development process [Luba88]. Methods that deal with reuse should be embedded within methodologies that support development, and environments to support reuse should be extensions of environments to support the developmental methodologies. This means that support for reuse must be an option supported and integrated into the standard mode of operation of the tool. The tool should support the storage, retrieval, and incorporation of reusable software with the new developed ones.

The difficulties and issues that arises to realize such a system can be summarized as follows:

- The amount of software is large and growing. This makes the search of software components difficult.
- Standards for writing and documenting software are not used or are incompatible from project to project.
- Systems are not designed for reusability.
- Various systems exist in many different programming languages and their incompatibilities pose a barrier for reusability.
- Software under development generally requires modification of reusable component, most intimate analysis of retrieved and developed component is necessary.

Therefore, we may conclude that realization of software reuse will require:

- Standards for specification, coding and testing that supports reusability.
- A model for reusable components that represents the features related to reusability.
- A design methodology such that reuse should be a continuing aspect of software development, embedded within the tool as an integral part of the methodology, rather than a separate option.
- An efficient and fast location, retrieval method from the software base. This should be a tool managing the database of component descriptions.

- Incorporation of the components ensuring that the combined components are fully semantic and context compatible with the program they are used in.

Computer Aided Prototyping System (CAPS) is designed to realize all these aspects. The requirement for the necessary standards for specification and representation is achieved through the use of Prototype System Description Language (PSDL). A software base equipped with an efficient search and retrieval mechanism is an embedded tool in CAPS. The retrieved components are incorporated into the prototyped system ensuring the fully semantic and context compatibility.

B. RAPID PROTOTYPING

The demand for large, high quality systems has increased to the point where a jump in software technology is needed. Rapid prototyping is one of the most promising approaches to this problem [BL88].

A prototype is an executable model of the proposed system. The prototype tries to realize the requirements such that the behavior of the proposed system is observable. This provides continuous modification of requirements as customers can visualize and test them throughout the development.

More than half of the total software cost originates from the changes made on software system, called maintenance phase in traditional software life cycle. Moreover, requirements errors are the most expensive errors to recover as major changes should be made on the system. Requirements are difficult to construct and validate because usually there is no single person who understands all of the constraints of proposed system. This is especially evident in large systems with hard real-time constraints, since the requirements for such systems are generally very difficult to understand or describe [Luqi88]. The communication problems between people with different areas of expertise worsens the situation.

Prototyping helps reduce maintenance costs primarily by reducing requirement errors. As prototyping allows extensive testing of a system, it also reduces the implementation errors.

Prototyping life cycle is in figure 1.1. The requirements are specified repeatedly by evaluating the executable prototypes and negotiating with the customer. The prototype is constructed based on the initial requirements. This prototype is a partial representation of the system which is used to analyze and design the system. Then, the designer examines the execution of the prototype together with the customer to compare the actual behavior of the prototype with the expected behavior. The feedback from the customer is used to adjust the requirements and to modify the prototype accordingly.

The repeated process of user validation and prototype construction continues until the prototype successfully meets all the aspects of the envisioned system. After that, the designer uses the validated requirements as a basis for the design of the production software.

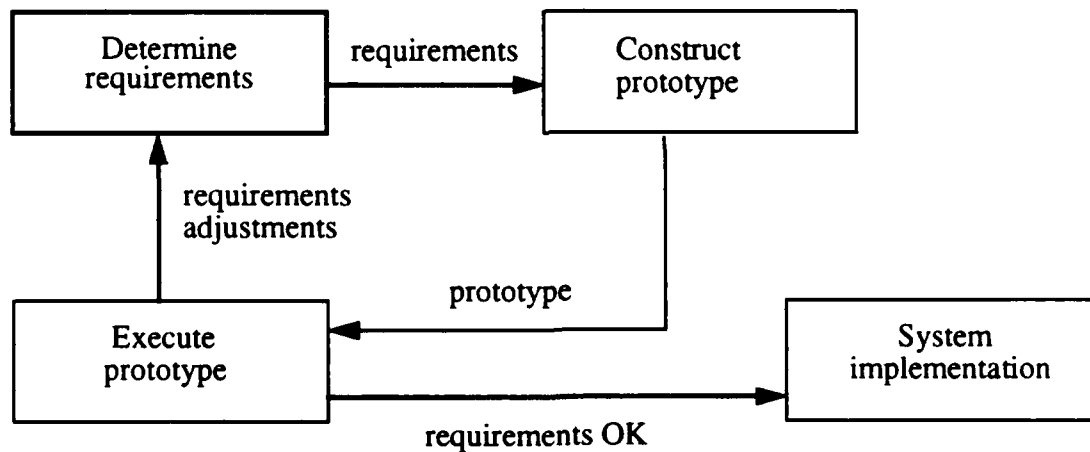


Figure 1.1 - Prototyping life cycle

The process of creating and modifying prototypes must be rapid enough for prototyping to be an efficient methodology.

We need software tools to make rapid prototyping practical. Reusing existing system components appears to be the most economical approach for constructing prototypes.

These reusable components should also have sufficient quality to obtain the necessary performance from the final system.

The method used for rapid prototyping should be supported by a clear, simple, and expressive computational model that is also supported by a matching language and automated prototyping environment. The same language must be used for prototype design and for software base retrievals to gain the benefits of reusable software components.

This model and language, namely CAPS and PSDL, are briefly described in the following two sections.

C. THE COMPUTER AIDED PROTOTYPING SYSTEM (CAPS)

The Computer Aided Prototyping System (CAPS) is an integrated environment aimed at rapidly prototyping hard real-time embedded systems [LK88, Luqi91]. CAPS software tools includes an execution support system, a syntax directed editor with graphic capabilities, a software base and an engineering database management system. Figure 2.2 shows the structure of CAPS.

CAPS provides a systematic design method for rapid prototype construction. CAPS prototypes a system through translation of the high level specification language Prototyping System Description Language (PSDL) [BLY88] into Ada code by making use of the stored reusable components. To generate a prototype, the designer of the prototype uses the graphic editor to create a graphic representation of the proposed system. This graphic representation is used to generate part of an executable description of the proposed system, represented in PSDL. PSDL specification will be used to perform an automated search of the component library for preexisting candidate implementations. An integration and transformation schema will then be used to transform the PSDL specification into the implementation language code that connects the retrieved reusable components. The prototype is then compiled and executed. The user evaluates the prototype against the

expected behavior of the proposed system. Repetitions of this process should yield a system that satisfies the requirements.

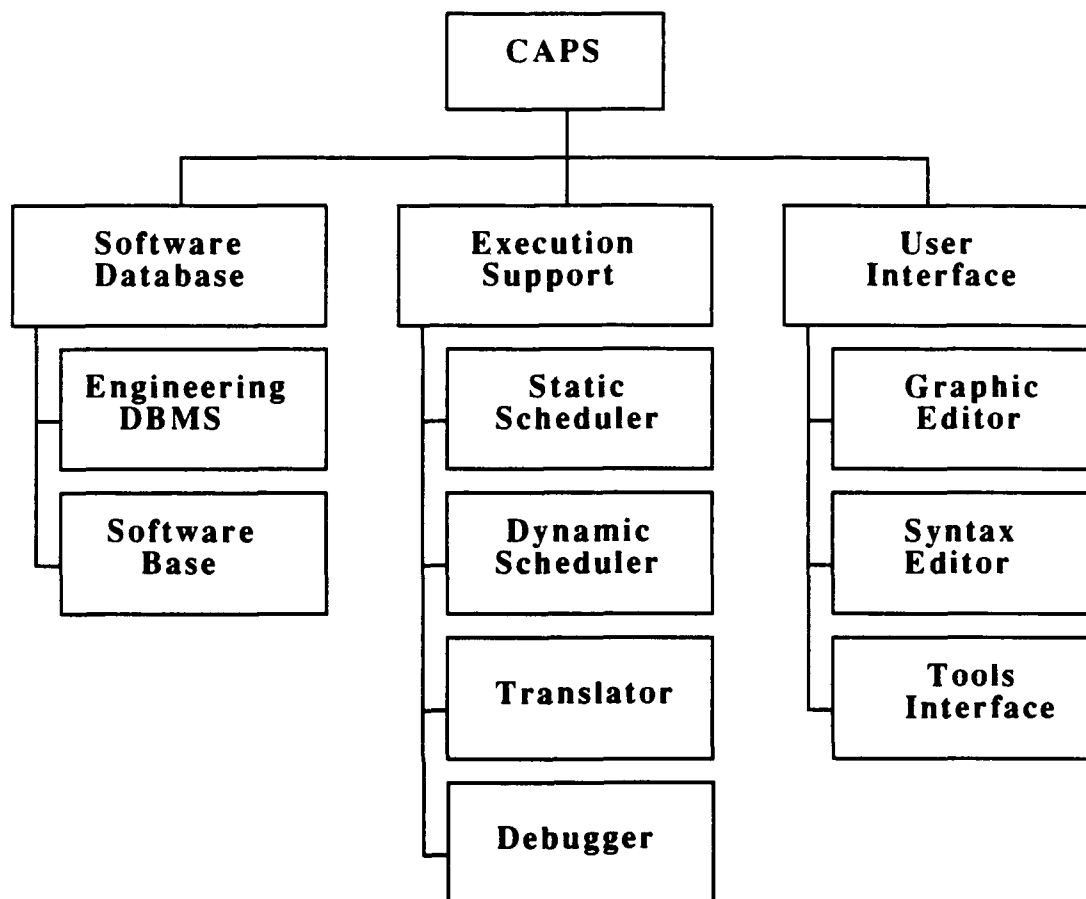


Figure 2.2 - Structure of CAPS

D. THE PROTOTYPE SYSTEM DESCRIPTION LANGUAGE (PSDL)

A powerful, easy to use, and portable prototype description language is a critical part of an automated rapid prototyping environment. Such a language is needed before the tools in the environment can be built.

PSDL [BLY88, BL88] is the specification language used in CAPS. It serves as an executable prototyping language at a specification or design level and has special features for real-time system design. PSDL has facilities for recording and enforcing timing

constraints, and for modeling the control aspects of real-time systems using nonprocedural control constraints, operator abstractions, and data abstractions. It is especially well suited for large real-time systems, and is useful for prototyping typical Ada applications. Ada is convenient for implementing PSDL because the mechanisms of Ada support the features of PSDL. Therefore, PSDL can easily interface to reusable Ada components as PSDL execution support system is also in Ada. PSDL is used to describe the connections between the components of a prototype, and to specify the behavior of the reusable components in the prototype as well as those in the software base.

PSDL provides two kinds of building blocks for prototypes: abstract data types and operators. Software systems are modeled as networks of operators communicating via data streams. The PSDL data types include built-in types of Ada, user defined abstract types, the special types time and exception, and the types that can be built using the type constructors of PSDL.

PSDL operators have two major parts: the specification and implementation. The specification part contains attributes describing the form of the interface, the timing characteristics, and both formal and informal descriptions of the observable behavior of the operator. Specification may optionally include inputs, outputs, exceptions, generic parameters, states and timing information. These interface characteristics are defined by the software engineer during the design process to specify the operator and form the basis for retrievals from a reusable component library or software base.

The implementation part determines if the operator is atomic or composite. Atomic operators have an implementation keyword specifying the underlying programming language, followed by the name of the implementation module implementing the operator. This name is supplied either as a result of a successful retrieval from the software base or is supplied by designer. Composite operators have the attributes communication graph, internal data, control constraints, and informal description.

E. OBJECTIVES

The major purpose of this thesis is to provide tools to facilitate the practice of software reuse within the Computer Aided Prototyping System (CAPS).

Currently, a software base provides prototype designers with the means to retrieve components. Reusable components are retrieved from the software base using a formal specification as the search key or through a browser. The specification language used is the Prototype System Description Language (PSDL). The software base stores the reusable components in an object oriented database management system (ONTOS) with an appropriate PSDL specification.

The study conducted in this thesis was aimed at achieving three major goals: (1) to modify the software base to allow the integration of the retrieved components into new applications; (2) to transform and integrate the chosen retrieved components to the system under development; (3) to design and develop a graphical user interface (implemented by using TAE Plus) for manipulating available software base operations, including the storage, retrieval, and integration of the components.

F. ORGANIZATION OF THE THESIS

Chapter II provides a brief background on information retrieval methodology and gives an overview of the current retrieval systems and tools. Chapter III describes the structure of the CAPS software base. Chapter IV discusses the integration of reusable components into new applications and provides the design methodology. Chapter V describes the design and implementation of software base graphical user interface. Chapter VI is the conclusions and recommendations for further research.

II. BACKGROUND AND PREVIOUS RESEARCH

This chapter describes some technical background concerning reusable software components and their retrieval, and reviews previous and current systems that implement various methodologies to achieve this purpose.

A. INFORMATION RETRIEVAL

The problem of retrieving reusable software components from a library is in general an information retrieval problem [Ste91]. The important concepts from information retrieval that relate to reusable component retrieval are representation, search, and measures of performance.

Representation is the way the object sought is structured to facilitate retrieval. For example a list of keywords may be provided for the future retrieval of a certain component. The method of representation must support the method used to search for the object. A lot of research has been conducted for search mechanisms, especially in the areas of artificial intelligence and database management systems. The method of representation and the method of search work together to form a cohesive environment for information retrieval. The more refined and precise the method of representation, the easier the search mechanism becomes. Various representation and search mechanisms will be described in the following section.

The two most important measures of performance are precision and recall [SM83]. Precision is the ratio between the number of relevant components retrieved and the total number retrieved. It is the answer to the question, "What percentage of the retrieved components are relevant?" Recall is the ratio between the number of relevant components retrieved and the number of relevant components in the database. It is the answer to the question, "What percentage of the relevant components on the database did my query find?" Precision and recall obtain ideal values when the set of components retrieved is exactly the same as the set of components that are relevant.

B. METHODS FOR RETRIEVING REUSABLE COMPONENTS

Almost all of the tools developed for retrieving reusable components use one or more of three different approaches for retrieval; browsers, informal specifications, or formal specifications. The fundamentals of each approach will be described in this section and then some existing tools using these methods will be reviewed in section II.C.

1. Browsers

A browser is a general purpose, usually window-based tool for looking through collections, categories, or hierarchies of components at various levels of abstraction [Meye88]. The objective is to allow the system user to manually search for the desired component.

The advantage of a browser is that the user is given complete control over the entire collection of components. This is especially useful if the user is familiar with the content of the software collection and already knows the components being searched.

The first disadvantage is that the method is basically manual, and the user should browse through the whole set of components to find the one that is desired. This also means a very low precision.

Second, it requires the user's knowledge of the structure of the component collection. Another issue related to this problem is that the search may be local, causing the failure or inefficiency of the retrieval; that is, if a suitable component is defined elsewhere in the system, it is not going to be found if user doesn't know to look at there.

Third, unless retrieved component is equipped with proper documentation, the user should look into the source code to determine if the component satisfy the requirements. Another related issue is that if the user has not found the exactly matching component for the needs, there will be no clear termination point for search.

Fourth, if the size of the software base is large, this method will be unacceptably insufficient and time consuming.

2. Informal Specifications

Retrieval techniques based on informal specifications require the user to describe or list some of the attributes of the component being looked for. Informal specification methods include keyword search, multi-attribute search, and natural language interfaces.

a. Keyword Search

Keyword search mechanisms require the user to specify a list of words relevant to the object being sought. Keywords can be drawn from a known system vocabulary (controlled vocabulary), or they can be unconstrained (uncontrolled vocabulary). In the case of unconstrained keywords, synonym tables are often used to find more standard words on which to perform the query [SM83].

The advantages of this approach are easy implementation and its conceptual simplicity for the user. These aspects make it a very commonly used technique.

One disadvantage of this approach is precision and recall of the system depend on the number of the keywords used. A lower number of keywords, for example, one, results in high recall and low precision as a large number of components will be retrieved; whereas too many keywords will have the opposite effect.

The second disadvantage is the user should be familiar with the structure of keyword categories. Both of this disadvantages cause the search to be an exercise of trial and error.

b. Multi-attribute Search

Multi-attribute search mechanisms are similar to keyword search mechanisms, but also use other characteristics of the component being searched. These characteristics that can be utilized for retrieval are the class of the object (procedure, function, package, etc.), the number and types of parameters, the number of operations it supports, its domain of use, etc.

The advantage of this method is that a component description contains more than just keyword information. The attributes taken together make up a classification scheme that provides more information than would be present in a pure keyword search.

The disadvantage of multi-attribute search is that the classification and subsequent storage location of a component defined by its attributes is left to the administrator, but users will not necessarily classify the same component the same way. If the user succeeds in filling in the same values, the query will be successful, otherwise recall of similar components will be low.

c. Natural Language Interfaces

This is a growing area of computer science research. An advantage of this system is the ease of language query formulations by system users.

In reusable component retrieval, the query will be issued through a natural language. Because of the ambiguity inherent in the broad semantics of natural language, a complex and detailed processing of the input language needs to be applied. Overcoming the difficulties of semantic analysis is the main challenge in dealing with natural languages.

3. Formal Specifications

There are many specification languages that has been used to describe the semantics of software processes. Examples of existing specification formalisms are predicate calculus [RW90b], plan calculus [RW90a], and algebraic formalisms [GTW78].

Because these specification languages are based on such specification formalisms they are free from ambiguity and can be transformed into normal representations by using logic and term rewriting rules without changing their meaning.

The disadvantage is specifications may be difficult for to write. Another disadvantage is that processing times for the search algorithms may be excessive depending on the approach taken. Finally, matching formal specifications is a hard problem.

C. CURRENT RETRIEVAL SYSTEMS AND TOOLS

This section describes some of the systems that have been built to perform reusable component retrieval and investigates the methods used by each system.

1. Draco

The Draco project [Neig84] is an approach to software engineering that has had a large impact on software reusability in general. The Draco approach focuses on domain engineering of software.

The most important aspect of Draco is the domain language. Software components are organized into problem areas or domains and a domain language describes objects and operations of a particular domain. There is a reusable component associated with each domain language object or operation. Since there is a potentially large number of components within a domain, a classification scheme is developed for the components called faceted classification to aid in organizing and retrieving the components.

Using faceted classification, Draco approach utilizes a multi-attribute query method. Queries are constructed by the formulation of a tuple of attributes that best characterizes a particular domain. A query session begins with the most specific query, that is, all attributes filled in. If the results of the query are unsatisfactory, the user may generalize the query by inserting wildcards for attribute values.

The advantages of faceted classification are that it is conceptually simple for users and relatively easy to implement. Because of this, the concept has been borrowed to implement the retrieval mechanisms in both RAPID (see Section II.C.2) and OSS (see Section II.C.5).

One disadvantage of multi-attribute search is that semantically similar components may not be found when their attribute definitions are different. Draco alleviates this problem by maintaining a measure of conceptual closeness for the term lists of each attribute. This way, an unsuccessful search can be tried again using an alternative but similar term in one of the attributes.

The disadvantages of faceted classification are that it is not suitable for unconstrained domains and semantically similar components may be missed from other domains, even with a conceptual closeness measure.

2. RAPID

The RAPID (Reusable Ada Packages for Information System Development) project is an ongoing effort in the Department of Defence [Voge90]. The objective of RAPID is to provide software engineers with quick access to reusable Ada packages in the information systems domain. The functions it performs are reusable software component classification, storage, and retrieval.

RAPID uses a faceted classification scheme to organize and retrieve components and falls into the category of multi-attribute search. No measures of performance or quality assessments are available yet.

3. The Reusable Software Library

The Reusable Software Library is a system designed to make software reuse an integral part of the software development process [BW87]. The system couples a passive software database with interactive software design tools to help software developers find and evaluate components to meet their requirements.

Components are stored in the database with attribute values that provide a basis for search. There are two methods available to search for components, standard multi-attribute search and natural language. The multi-attribute approach provides a menu driven interface in which the user selects the attributes with which to perform the search. Alternatively, the user may express his query in the form of natural language, such as "I need a stack package.". The system parses the input, extracts keywords from it and uses those words as attributes to perform the search.

The designers of the system report that the natural language front end is considerably easier to use but the search is significantly slower, by a factor of five to ten because of the natural language parsing overhead involved.

4. The Programmer's Apprentice

The goal of Programmer's Apprentice project is to apply artificial intelligence techniques in an effort to automate the programming process [RW90a]. It is designed to provide intelligent assistance in all phases of a programming task.

A reusable component is called a cliché. A cliché represents a commonly used combination of elements such as abstract data types, binary searches, etc. Thus, programs may be considered as collections of interrelated clichés.

A formalism called *Plan Calculus* is developed to represent clichés. A plan defines a single cliché in three parts: a plan diagram, a logical annotation, and an overlay. Together these parts constitute a language independent formalism for describing reusable software components.

A maintenance tool called the *Recognizer* automatically finds all occurrences of a given set of clichés in a program and builds a hierarchical description of the program in terms of the clichés found. It is not clear whether the *Recognizer* will be used as a general purpose component retrieval tool. It is currently limited to finding algorithmic clichés but the researchers hope to extend its capability to find data structures and data abstraction as well.

5. Operation Support System

The Operation Support System (OSS) is an ongoing project aimed at developing an integrated software engineering environment undertaken by Naval Ocean Systems Center [Ste91]. One goal of the project is to establish a Navy software library of reusable software components.

The current prototype library subsystem allows component retrieval using faceted classification, keywords, or a textual browser. The components currently stored in the library are large command, control, and communications subsystems. Since the library is in its early stages, no information is available on its performance characteristics.

6. Specifications as Search Keys

An experimental system developed at Carnegie Mellon University uses formal specifications to search software libraries [RW90b]. Each function in the library has a corresponding formal specification. Specification matching is the process of determining whether a specification of a library function satisfies a query. Specifications and queries are written in λ Prolog.

Each specification has a signature and some semantic information. Their aim is to match first on signature and then increase precision by matching on specification semantics. The system designers claim that the use of semantics in specification matching increases precision. Although they give examples showing precision is improved, they do not provide statistics that indicate how much.

III. CAPS SOFTWARE BASE

The paradigm for rapid prototype construction in CAPS leads the designer from a graphical representation of the prototype, through specification with a prototyping language (PSDL), and then on to code generation. Figure 3.1 shows the prototyping process supported by CAPS [LK88].

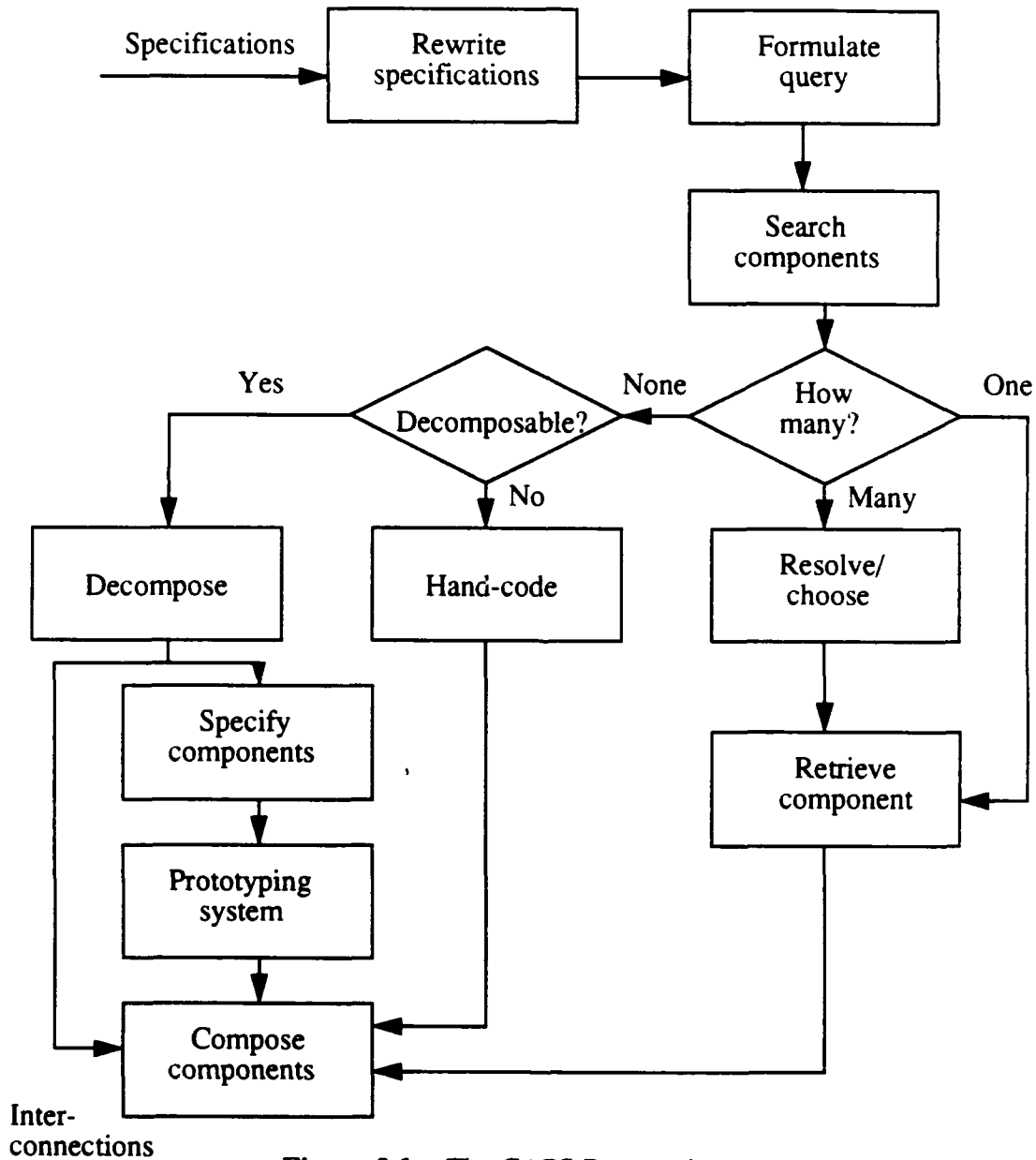


Figure 3.1 - The CAPS Prototyping process

As seen, CAPS takes advantage of a library of reusable software components. The prototype designer writes specifications for the operators and data streams to model system requirements, these specifications are used to locate components that will satisfy those requirements. A retrieval system that is automatic, efficient, and effective relieves the designer from having to use a browser or some other manual means to locate components. This is particularly beneficial when the software base contains a large number of components. Once the desired component is retrieved, the next challenge is to integrate it into the system.

The functionality of software base can be divided into three main categories: Storage, retrieval, and integration of components. This scheme can be viewed in figure 3.2.

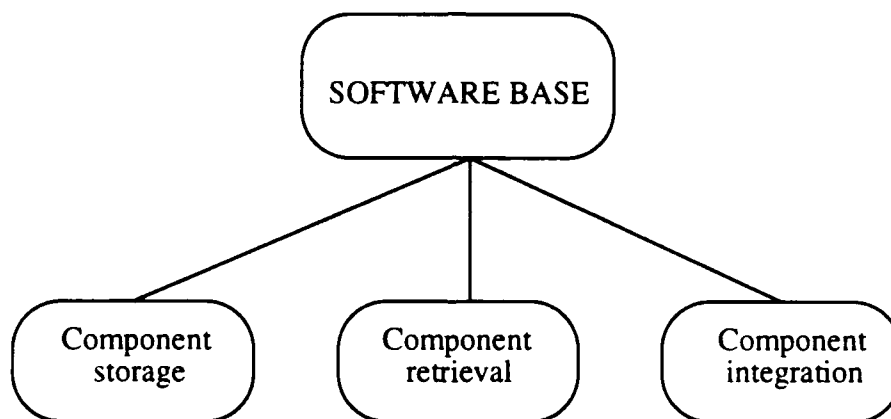


Figure 3.2 - Functionality of Software Base

The storage and retrieval facility of software base is described in the remainder of this chapter, and integration process is described in chapter iv.

A. COMPONENT STORAGE

Due to the complexity of storing variable length source code and querying the software base using PSDL specifications, a powerful DBMS is necessary. This DBMS should also support multi-user and networked access to its data as CAPS is designed for multi-user

networking environment. This database should perform the component storage facility of the software base.

1. ONTOS Database Management System

Ontos is a distributed object database management system [Onto90]. It consists of a database, a set of utilities, and a C++ interface. It was selected for use in the software base project because it has sufficient capabilities to handle the requirements for the implementation of an advanced reusable software component library.

Ontos allows the database developer the ability to make any data object persist past the execution of the program that created it. To achieve this, the ONTOS class library introduces an Object class. It is the parent of all persistent classes. Object defines a constructor for creating objects in the database and a destructor for deleting them. The Object class constructor assigns each object a unique identifier and provides methods to store and retrieve them.

Ontos includes a set of persistent aggregate classes in order to efficiently handle collections of persistent objects. The aggregate classes include Dictionaries, Lists, Sets, and Arrays. The aggregate class hierarchy is shown in figure 3.3.

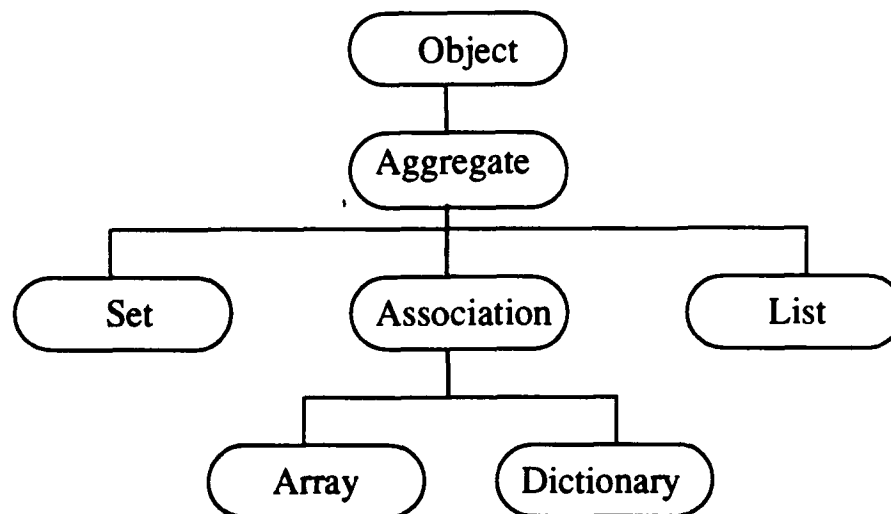


Figure 3.3 - The Aggregate Class Hierarchy

The aggregate classes are container classes and are used to deal with groups of objects. The List class is analogous to a linked list data structure. The Set class implements the standard concept of set and the Array class implements the concept of arrays. The Dictionary class is the most versatile of the aggregate classes and it is extensively used to implement the structure of the software base. All of the four classes are fully dynamic and can grow without bound.

The Dictionary class is a keyed data structure that can be ordered or unordered. Every entry in a Dictionary has two attributes stored, the Tag and the Element. The Tag is used for indexed look up and the Element is to hold the data. Dictionaries may allow or not allow duplicate elements for a given Tag.

Ordered Dictionaries are maintained in ascending key order and use B-tree access structure to sort entries based on the relative ordering of their tag values. Unordered Dictionaries use hash table access structure.

2. Storing The Components

The CAPS software base [McDo91] is designed as a *general purpose* tool capable of storing components implemented in many programming languages. Because of the differences in the capabilities of each programming language there are differences in the way the pre-defined abstract data types used in PSDL to specify components are interpreted by the software base. These differences in the interpretation of PSDL specifications require that all components of a particular implementation language be considered in a unique domain. It is also possible to create multiple component domains for a given implementation language. Each domain is a library composed of five parts: a component dictionary, a keyword library, an operator library, an abstract data type library, and a recognized type matrix.

For each component in a domain, six text files must be stored. These files are the PSDL specification, the implementation language specification, the implementation body,

the informal description, the axiomatic specification, and a normalized version of the axiomatic specification.

A persistent class was designed inheriting from the Ontos class Object to allow storage and retrieval of these text files. Instances of this class is used as attributes of each component in the software base to store the files.

B. RETRIEVING THE COMPONENTS

There are three methods to retrieve a stored component: by PSDL query, by keyword query, and through a component name browser. Although browsing by component name and keyword querying are not the preferred methods for finding reusable components in a large software base, they are necessary features of any software collection.

Each domain is divided into two categories: abstract data types and operators. An ordered list of components is provided for browsing the components by their name in these disjoint categories.

Each software base library also includes a keyword library for handling keyword access to its components. The keyword attribute of PSDL specification is used to perform a keyword structured search in the software base. The result of a keyword query is a list of those components that posses one or more of the query keywords. The list is ordered with those components that satisfy the most query keywords coming first.

1. Query by PSDL Specification

This method uses the syntax and semantics of the PSDL specification to conduct the search for a component. First, the PSDL specification is modified to improve the efficiency of the search. The process of transforming or manipulating the specification for a reusable software component is called normalization.

Components to be stored goes through a syntactic and semantic normalization process that transform the component's PSDL specification to facilitate later matching. The normalized specification is stored with the component in the software base. At the retrieval phase, a query for a library component is formed by constructing the PSDL specification

for the desired component. The query specification is syntactically and semantically normalized and then matched against the stored specifications.

Syntactic normalization standardizes the form of the query's interface characteristics to be used in syntactic matching. Semantic normalization transforms the signature and axioms of the specification to make them suitable for semantic matching [Ste91]. Syntactic matching takes place before the semantic matching. Syntactic matching is faster and quickly narrows the list of possible candidates while semantic matching is time consuming and should be applied to as small a candidate list as possible. The main benefit of syntactic matching is speed whereas the advantage of semantic matching is accuracy. This schema is shown in figure 3.4. The functional programming language OBJ3 is used for semantic matching and further details can be found in [Ste91].

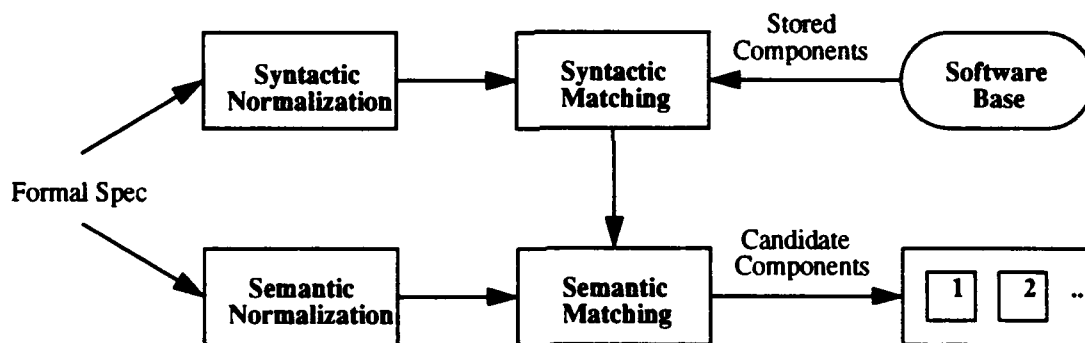


Figure 3.4 - Normalization for Component Retrieval

IV. INTEGRATION OF REUSABLE COMPONENTS

Once a reusable component is retrieved, incorporating it into the developed software system is an important task of a software base. Software base should support the automatic adaptation of the retrieved components into the program at construction time ensuring the full context compatibility. This process will require a transformation procedure to be applied on the retrieved components such as changing the parameters, operator names etc. as well as instantiating the generic parameters.

A. INTEGRATING THE REUSABLE COMPONENTS INTO CAPS

Software base should provide a component implementation that meets the needs of the CAPS execution support system. To accomplish this, the retrieved component should be transformed into a suitable form to be used by the implemented system. On the other hand, software base cannot directly generate implementation code as it is not language specific. Therefore, we need an integration tool to generate the necessary code to incorporate the component into the system. In the rest of this chapter, the methodology and the implementation of an integration tool developed for incorporating the retrieved components into the CAPS is discussed.

1. An Integration Tool For CAPS Software Base

The software base provides three methods to retrieve a stored component as discussed previously. The kind of method used for retrieval is important for incorporation. We have a PSDL query containing the requirements of execution support system if component is retrieved by query, and no specification is provided if keyword query or browsing by component name methods are used.

If component is retrieved by PSDL query, we can use this query specification in transformation to change the input/output parameter names, operator or type names, and to instantiate the generic parameters if there is any.

If retrieval is conducted through other methods, no change of parameter and operator/type name is necessary, but we still should have means to instantiate the generic parameters and transform the retrieved component into a form that can be used by the system.

B. METHODOLOGY FOR TRANSFORMATION

It has been difficult to reuse software components in practice as in most cases two instances of the same component are rarely alike. Typically, many small variations on a theme exist and generally it is difficult to predict which variation will be needed in a certain implementation.

This situation can be alleviated by providing a transformation mechanism that adapts the components explicitly stored in software base. This capability has two important effects on software reuse. First, retrieval process has a much better chance of successfully retrieving a specified component from the software base compared to a system that can only return components explicitly stored in the software base. Second, automatically adapting the reusable components after retrieval is important in rapid prototyping where designer time is at a premium.

Transformations should take place in such a way that it should have no effect on the stored component, yet it should be readily usable by the system. This means that instead of modifying the retrieved component, it should be used as a basis for the transformation module. In our case, this is accomplished through the *with* statement for Ada libraries. The same schema may be applied to other languages through an inheritance mechanism as it will be the case in C++. This module should be automatically generated as a part of the retrieval/integration system.

In our application, the transformation module will be created for each retrieved component and will be embodied in its own package through the integration process. This package will be saved in the work area and will be used by the designer. All name and parameter changes as well as generic instantiations will be embedded in this package.

Because there are various ways to retrieve a component, the creation of this transformation module varies for each method.

The implementation of transformation module is discussed in sections 4.D and 4.E. The structure of integration tool is discussed in the next section.

C. STRUCTURE OF THE INTEGRATION TOOL

The Integration Tool is used to:

- (i) input PSDL of the retrieved component (and query if PSDL query method is used),
- (ii) transform the required component, parameter names and perform instantiations,
- (iii) output the transformation module encapsulated in its own package.

The structure of the integration tool is in figure 4.1.

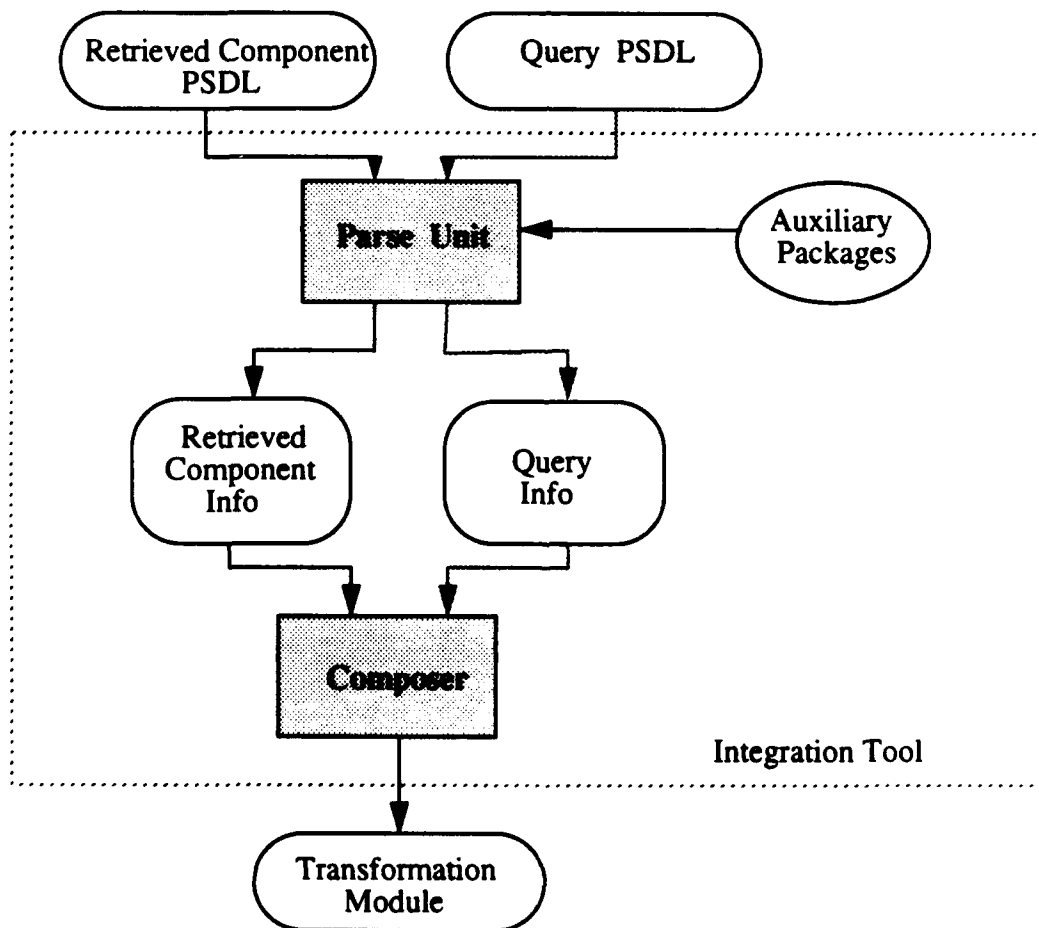


Figure 4.1 - Structure of The Integration Tool

In the first step, the PSDL specification of the retrieved component and the PSDL query (if the query by PSDL specification method is used for retrieval) are parsed by a LARL(1) parser. This parser is constructed by using the tools *ayacc* and *aflex*, which are Ada versions of the parser generator tools *yacc* and *lex* that are provided in UNIX environment [TTS88, Self90].

In the second step, PSDL of the retrieved component is evaluated. Depending on the retrieval method, various structures that will be used to compose the transformation module are created. If retrieval is performed by PSDL query, operator and/or type names and parameters of retrieved component are also mapped to the corresponding descriptions given in query.

In the third step, an Ada package is created and the transformed representation of the retrieved component is written into this package to be used by the prototype.

1. PSDL Parser

Two LARL(1) parsers are used to parse the PSDL specification of the retrieved component, and the PSDL query if supplied. Because the actions and the resulting structures to be formed will be different for two types of integration process we need two different parsers, one for the integration of the components retrieved by PSDL query, one for the integration of the components retrieved by named look up or keyword query. Each parser is constructed by using a parser generator -*ayacc*, and a lexical analyzer -*aflex*.

Two auxiliary packages are supplied to provide the structures in which the information about the parsed PSDL specifications will be stored for use during the composing process. These packages also contain the Ada code for the semantic actions which will take place during the parsing process of PSDL specifications.

The top level diagram of the parser is in Figure 4.2.

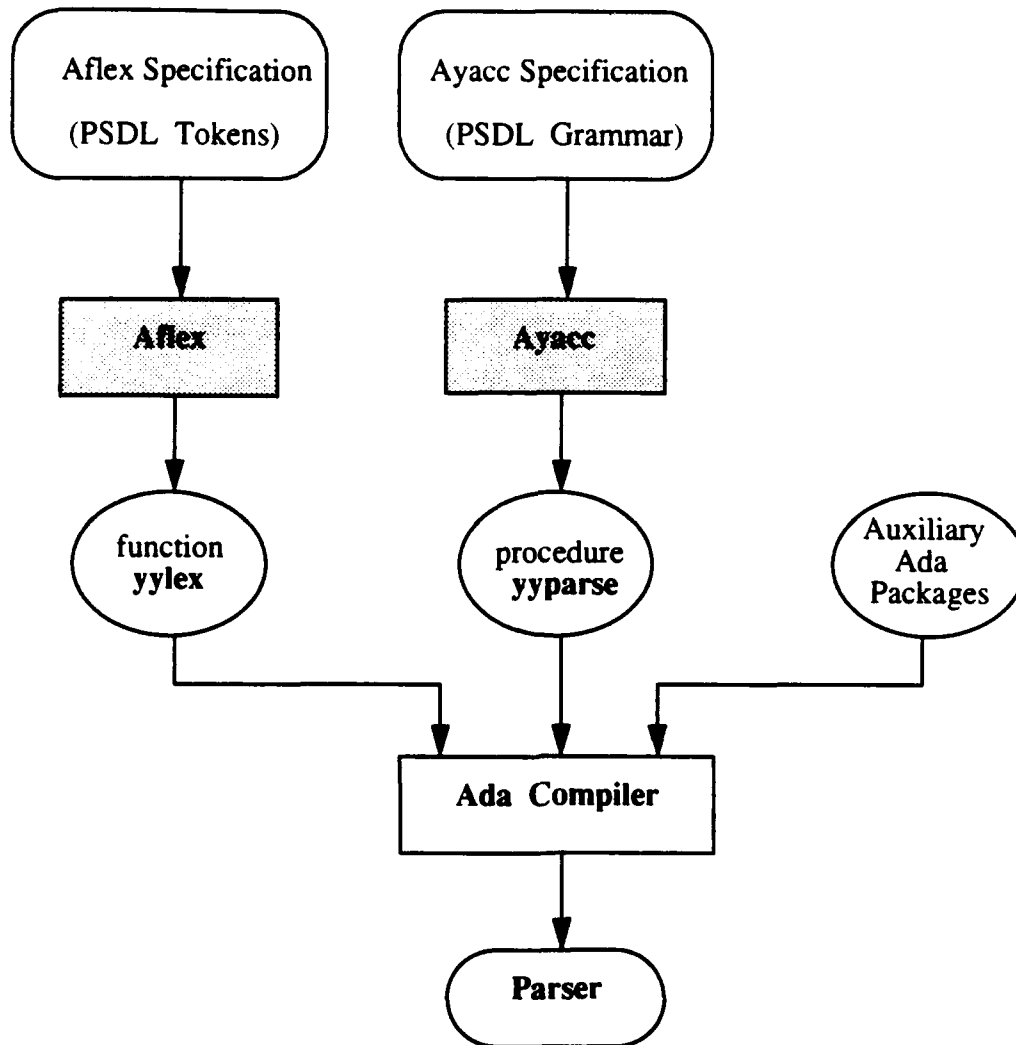


Figure 4.2 - Parser Generation

a. Parser Generator (Ayacc)

Ayacc [TTS88] generates a parser from an input of BNF style specification grammar, accompanied by a set of Ada program fragments called actions to be executed as each grammar rule is recognized. The generated output is a set of Ada program units that act as a parser for the input grammar.

The main parser procedure *yyparse* makes a call to lexical analyzer function *yylex* to get an input token, and then matches the grammar rules and executes the actions associated with these grammar rules. Ayacc needs a specification file as input which is a collection of grammar rules and actions associated with them, along with the Ada subprograms we provided to be used in the semantic actions. The input specification files provided are *psdl.y* and *query.y* containing the rules to process the PSDL specifications of the retrieved component and query respectively.

Ayacc generates four files, three auxiliary packages and a parser. The parsers generated for our application are in the files *psdl.a* and *query.a*.

b. Lexical Analyzer (Aflex)

Aflex [Self90] is a lexical analyzer generating tool written in Ada designed for lexical processing of character input streams. Aflex accepts high level rules written in regular expressions for character string matching, and generates Ada source code for a lexical analyzer, by using a finite state machine to recognize input tokens. Aflex can be used alone for simple lexical analysis, or with ayacc to generate a parser front-end, as in our application.

Aflex generates a file containing a lexical analyzer function (*yylex*) along with two other auxiliary packages. Lexical analyzer is implemented as an Ada package in this application. The files *query_lex.l*, for retrieval by PSDL query, and *psdl_lex.l*, for other retrieval methods, are the inputs to aflex. They define the lexical classes and the regular expressions used in the PSDL grammar. Each regular expression is associated with an action which is written in Ada and executed when the regular expression is matched. The lexical analyzer function *yylex* returns a single token after each call, made by the parser procedure *yyparse* of ayacc. The name of the generated file is *psdl_lex.a*.

2. Transformation Process

Retrieved component is used as a basis for the creation of the transformation module. This process is conducted in two phase. First, the PSDL specification of the

retrieved component is parsed and the necessary information about the component is stored in the structures supplied in the auxiliary package *component_aux_pkg*. This information includes the name of the component, whether the component is a type or operator, generic parameters, and input/output parameter names and values. If the component is a type, information related to each operator is also extracted during the parsing process.

Second, the retrieval method is evaluated in the composition phase. If retrieval process is accomplished through named look up or keyword query, the creation of the transformation module is conducted by using the information extracted from the PSDL specification of the component during parsing. This process is analogous to the use-as-is methodology. Although it is not the preferable method of code reuse, it can still be utilized. Therefore, we included an integration mechanism for these kind of retrieval and reuse schemes as well. Output of composer is the transformation module containing the implementation that incorporates the retrieved component as is through Ada with statement and renaming. This procedure is explained in detail in section 4.C.2.a.

If retrieval of the component is accomplished by PSDL query, we need a mapping mechanism. For this purpose, the PSDL query is also parsed and similarly the information about the query is stored in a separate structure supplied by the auxiliary package called *query_aux_pkg*. This information is used to transform the component name, operator names (if component is a type), and parameter name and values to maintain the consistency in the system.

Mapping process is carried out during the composition of the transformation module by using the information from both structure packages, namely *component_aux_pkg* and *query_aux_pkg*. During this mapping procedure generic parameters are also taken into consideration. If retrieved component is generic, the value of each input and output parameter is checked against the generic parameters. This issue is addressed in section 4.C.2.b.

The resulting transformation module is implemented as a package and saved in the prototype directory. Thus, implementation for the integration of each retrieved

component is encapsulated in its own package. The file this package is written into is called *proto-prefix.atomic.a* where *atomic* stands for the component name. The details about the creation of the transformation module is in the following two sections.

a. Transformation Module for The Components Retrieved by Browsing

Keyword query and named look up of components are two methods used for reusable component retrieval. None of these methods provide the specifications required by the design process. Nevertheless, we should provide ways to integrate a component retrieved by these methods as it may be desirable to use this component as it is. In this case no parameter change is required. We need to generate a package *witned* to the retrieved component and rename the contained procedures.

For this process, the PSDL specification of the retrieved component is used to generate the transformation module. Figure 4.3 is the schema for the creation of transformation module.

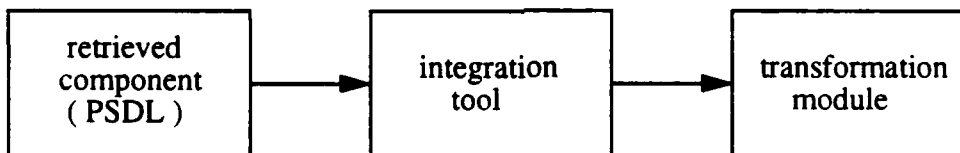


Figure 4.3 - Integration process

The PSDL specification of the retrieved component is parsed and the values to construct the transformation module are extracted. These values, which are component name, parameter name and values etc., are then used to construct the transformation module. At this phase, the generic parameter values should also be supplied to perform the instantiation of generic components. A detailed discussion of generic instantiation can be found in section 4.C.3.

After this process, an Ada package containing the transformation module is created and written in the *proto-prefix.atomic.a* file by utilizing the package *component_compose_pkg*. Resulting package is saved in the prototyping directory.

An example template about the contents of the generated package can be seen in Figure 4.4. ADTs and procedures (corresponding types and operators in PSDL specification) are both incorporated the same way.

proto-prefix.retrieved_component.a

```
with retrieved_component;
package retrieved_component_pkg is
--if generic
    package tmp_retrieved_component_pkg is new
        retrieved_component ( -parameters- );
    procedure p1 ( -parameters- ) renames tmp_retrieved_component_pkg.p1;
    procedure p2 ( -parameters- ) renames tmp_retrieved_component_pkg.p2;
end retrieved_component_pkg;
```

Figure 4.4 - Generated package of transformation module

b. Transformation Module for The Components Retrieved by Query

Designer will create a PSDL specification, which will describe the properties of the desired component, during the prototyping process and will use this specification to query the software base to find a matching component. If there is any matching component in the software base, the retrieval will be successful and a list of retrieved components (one or more) will be available. Then the designer will choose the component which best suits to his or her design and will integrate that component into his/her system.

The PSDL query contains the specific information about the required component such as the desired component name, parameter name and values, procedure names etc. Although the retrieved component satisfies the query, it will probably have different component name, parameter name and values, and may even be generic. In this case we need to transform these values and names, and instantiate the component if it is generic.

Integrating such a component requires a mapping schema. The names and values of the retrieved component should be mapped to those of the names and values of the query. This process is accomplished by the integration tool and the resulting transformation module contains the implementation consistent with the query. Figure 4.5 gives the top view of the schema to perform this operation.

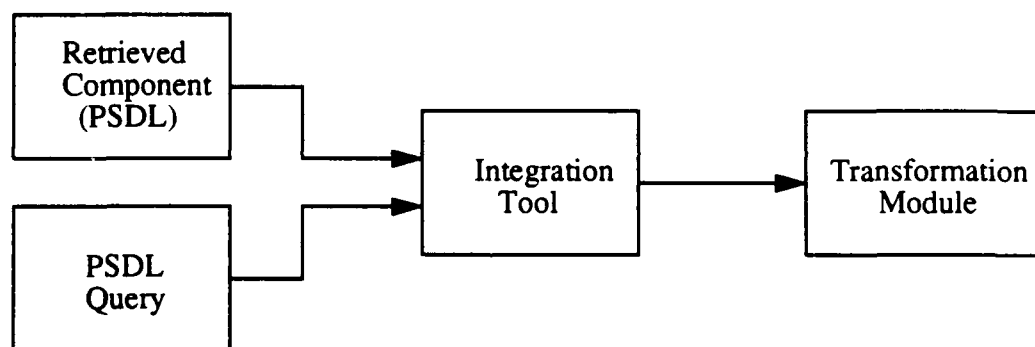


Figure 4.5 - Integration Process

The PSDL of the retrieved component and query PSDL are parsed and portions of relevant information is stored in respective structures. During the composing operation the retrieved component is *with'ed* to the transformation module. If the retrieved component is generic, it is instantiated with the generic values supplied. Then, depending on whether the retrieved component is operator or type, each procedure is renamed according to the procedure and input/output parameter names in the query PSDL.

Meanwhile, each input/output parameter value is also checked to determine whether it is generic.

After this procedure is carried out, the implementation is consistent with the PSDL query. The transformed representation of the retrieved component is then saved into the prototyping directory and it is ready to be used in the prototype. The name of the file the implementation is saved in is *proto-prefix.atomic.a*, where *atomic* stands for the desired component name specified in the PSDL query file. Figure 4.6 is an example output for the transformation module, *query_component* stands for the name of the component specified in the PSDL query.

proto-prefix.query_component.a

```
with retrieved_component;
package query_component_pkg is
--if generic
    package tmp_query_component_pkg is new retrieved_component(-parameters-);
    procedure query_p1 (-parameters-) renames tmp_query_component_pkg.retrieved_p1;
    procedure query_p2 (-parameters-) renames tmp_query_component_pkg.retrieved_p2;
end query_component_pkg;
```

Figure 4.6 - Generated package of transformation module

One problem that may be encountered during this process is the retrieved and query (desired) component names being the same. In this case renaming the retrieved component with the same name will cause an error with respect to Ada rules. To prevent this problem a certain naming convention must be applied.

One solution is appending a suffix to the component name given in the PSDL query, but this will also require a change in the PSDL query to maintain the consistency between the query and implementation. A better solution is to store every component in the software base with a known suffix. Thus, their corresponding PSDL specifications will be consistent and will require no further modification. The suffix used in this implementation is *_sb*, i.e. the name of a sort procedure stored in software base will be *sort_sb*.

A detailed example of integration process is given in Appendix A.

3. Generic Instantiation

In case the retrieved component has generic parameters, instantiation of the component during the transformation process should take place. There are a few aspects that should be taken into consideration during this process.

First, we need to be able to instantiate the generic components retrieved by named look up or keyword query. In this case, the generic values are not available to the software base. Generic parameter values should be supplied by user.

If retrieval is accomplished through PSDL query, the query may contain some or all of the generic parameter values. The challenge here is to match the generic parameters with the correct values supplied by the query. For example, if two generic parameters exist in the retrieved component and two values are supplied by the query PSDL, during instantiation correct values should be used for the corresponding appropriate parameter. This situation is worsened if the supplied values by the query PSDL are user defined type. In this case a chain mapping of generic parameter values will be required. These user defined types may even be defined in another PSDL specification as this is a common practice in Ada programming where type definitions are generally encapsulated in a separate package. In this case the required PSDL file should be searched and appropriate generic values should be extracted from this file.

Considering the difficulties of such an implementation, an alternative way may be to redefine the user defined types in the query PSDL and/or to modify the query such that all generic values can be extracted and matched to the generic parameters.

In this implementation the transformation module generates the template for the instantiation of the generic parameters and a message is displayed to the user to request the generic parameter values.

V. GRAPHICAL USER INTERFACE FOR SOFTWARE BASE

The software base graphical user interface (GUI) is designed to demonstrate and manipulate the capabilities of the software base as efficiently as possible by applying generally accepted user interface design principles. This chapter outlines the software base graphical user interface design decisions. Appendix B is the user's manual, and Appendix D contains the source code for the software base graphical user interface.

A. DESIGN PRINCIPLES

The key goals of a user interface design are to make the system easy to learn and use, to reduce the error rate, and to be intuitive to the user. This requires a user interface to be consistent, provide feedback, minimize error possibilities and provide error recovery, accommodate multiple skill levels, and minimize memorization.

To satisfy all these goals is not easy in practice, and requires a through analysis of both the potential users and the required functionality of the underlying system.

1. Users

a. Assumptions

The CAPS users are expected to be a large spectrum of people, a user can be a novice student or a highly skilled software engineer.

All users are assumed to be accustomed to graphical windowing environment. They are considered to know the basic skills such as using a mouse, working with pull-down menus or push buttons, etc.

The user is assumed to have a basic knowledge of PSDL specification language and its functionality in the CAPS. He/she is not expected to be an expert.

The user is assumed to be familiar with the implementation language he is working on. Currently all operator and types are implemented in Ada.

b. Requirements

Considering the assumptions about the user and preferred design decisions, an efficient user interface should satisfy these basic requirements for user:

- Must be suitable for a wide range of users from novice to expert.
- Must be easy to learn and use, should require no user manual for operation.
- Must use a graphical, mouse driven interface.
- Must provide on-line help.
- Must show user only relevant information and steps.
- Must display only one panel/menu item at a time.
- Must provide feedback to the user.
- Must give warnings if a destructive action is going to take place.

2. System

Since CAPS is an ongoing project at the Naval Postgraduate School, underlying system requirements are easy to determine. The Software base GUI:

- Must run on Sun or compatible workstations under Unix operating system.
- Must run under X-Windows for portability.

B. IMPLEMENTATION CONSIDERATIONS

The previous version of software base user interface was written using Stanford's InterViews toolkit (version 2.6) as a prototype. InterViews makes extensive use of C++ features such as inheritance and provides a number of predefined objects, including push-buttons, scrollbars, file management dialog boxes and text editors. Although it was a good candidate to implement this complete version of software base graphical user interface, the goal of CAPS is to implement the system tools in Ada as much as possible. Currently, the only existing tool that can be used in user interface design making use of Ada is Transportable Applications Environment (TAE) [Nasa91]. This aspect of TAE as well as the other qualifications described in the following section, has played a key role to choose the TAE as the toolkit to implement the software base graphical user interface.

1. Transportable Applications Environment (TAE)

a. Introduction

The software base graphical user interface is implemented by using Transportable Applications Environment (TAE) Plus, an integrated environment for developing and running contemporary window and graphics based application systems. TAE Plus is a portable software development environment that supports rapid building, tailoring, and management of graphic-oriented user interfaces. It provides a suite of integrated tools and software libraries for developing and running highly interactive, graphical application systems.

b. Environment

TAE Plus utilizes X window system, Version 11, Release 4, and the Open Software Foundation's Motif Toolkit, Version 1.1.1. InterViews 2.6 is also bundled with TAE Plus.

Figure 5.1 illustrates the relationship between a TAE Plus application and the X11 software.

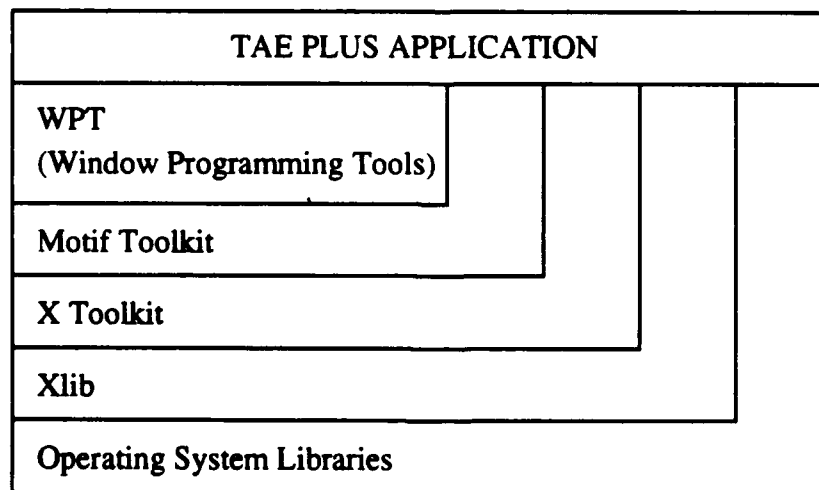


Figure 5.1 - TAE Plus, Motif, and X11

The application primarily calls Wpt functions. The Wpt package is layered on top of the Motif Toolkit, which interfaces with the X Window System. Motif Toolkit provides various high level interface objects called "widgets". X Toolkit is based on Xlib, which provides useful set of graphics primitives.

While the Wpt routines provide a convenient buffer between an application and the complexities of X11 and the X Toolkit, an application is free to access the X Toolkit and Xlib directly.

TAE Plus is portable across a wide range of computing platforms using either UNIX or VAX/VMS operating systems, and is capable of generating code in C, Fortran, and Ada.

c. Advantages

- Sophisticated GUIs can be built in less time.
- Easy to use.
- Flexible and modular, changes can be made locally without effecting the other parts of the system.
- Portable, it can be utilized on many different computing environments.
- Reusable, interaction objects and panels can be reused and shared avoiding the duplication of the effort.
- Consistent, it provides interfaces that have a common look and feel across different applications and environments.
- Possibility of implementation in different languages.

C. SOFTWARE BASE GUI ARCHITECTURE

The software base GUI is designed to manipulate software base operations through a series of user interactive panels. These operations can be classified into three category:

- (i) library operations,
- (ii) retrieval operations, and
- (iii) manipulation operations.

The software base GUI can be started by executing the command `softbase.exe`, by passing two command line arguments. First argument is the path to the prototype directory and the second argument is the prototype prefix, which is required by the software base script for compilation. The argument *prototype-directory-path* specifies the current prototype directory where the design process is carried out. The retrieved component and the integration package are saved in this directory as default. This path can also be entered as an environment variable (for example, `$PROTOTYPE`). The argument *prototype-prefix* is added to the beginning of the names of each saved file. This prefix is used in software base script to group each compilation unit.

Due to the current equipment setup in CAPS lab, it is necessary to invoke the software base on `sun51`, since Ontos libraries is on `sun51`.

Online help is provided for all major software base operations. Various warning and error messages are also included to help the user.

1. Library Operations

The panel for library functions is the top level of the software base GUI system. It provides selection, addition, and deletion operations to be performed on software base libraries.

If the creation of a new library is required, library addition panel provides input for the name of the new library and the rule file to be used. This operation allows one to create new libraries (for example `C++`, `Ada1`, `Ada2`, etc.).

Library deletion panel is used to remove an existing library and all of the components of that library. Library addition and deletion operations are accessible only by the authorized personnel (`root`).

Library selection option is provided for the user to perform his/her operations on the desired library. After selecting the library name, the rest of the operations are performed on that particular library.

2. Retrieval Operations

After the desired library is selected, a main menu panel provides access to various retrieval operations. This panel also provides the option of component addition to the chosen library.

The main menu panel provides options to conduct a PSDL or keyword query. A PSDL query file can be selected by browsing through the directories and it is used to perform the query. For keyword query, available keywords are displayed and the user can select as many keywords as he/she desires to conduct the query. The result of the queries is displayed in a separate panel and the user may select the most suitable component to suit his/her needs.

This panel also provides the named look up of components under two categories: operators and types. User can choose a component by browsing through the existing components although this would not be an efficient way of retrieval if that library contains a large number of components.

3. Manipulation Operations

Following the retrieval of a component, various operations can be performed for further manipulation. One of the operations is to view the files stored in the software base. These files are the PSDL specification, the implementation specification, and the implementation body files.

Other available operations are to delete the selected component from the software base, to save the component to the workplace, to print the specification and implementation files of the component, and to integrate the component into the system under development.

D. CONCLUSIONS

TAE Plus is an efficient tool to design and implement sophisticated graphical user interfaces. It is also suitable for hardware platforms that CAPS runs. User interfaces consistent with the other CAPS tools user interfaces can easily be developed by using TAE

Plus. That Ada is one of the implementation languages of this tool is also suitable, given that one of the major goals is to implement the whole CAPS in Ada.

Evaluating these benefits of TAE Plus, the decision is made to use this tool to implement the software base graphical user interface. The developed interface is a mouse driven, easy to use graphical user interface which is consistent with the requirements given in section 5.A. On-line help facilities are also included, although it is pretty straight forward to use the system. Various panels for information, warning and error messages are also included.

The software base graphical user interface user's manual is given in Appendix B, explaining the details of the developed system.

VI. CONCLUSIONS AND FUTURE RESEARCH

A. ACCOMPLISHMENTS

Three basic tasks are accomplished in this thesis to enhance the software reuse within the CAPS.

First, the software base system first described in [McDo91] was revised and modified to accommodate the requirements of current CAPS system. Major changes made were in parser and keyword query modules. Parser was modified according to the requirements of current PSDL specification rules. Keyword query module was changed to allow multiple keywords to be used as search keys.

Second, an integration mechanism has been designed and implemented to incorporate the retrieved components into CAPS. An integration module is created and saved to the workplace for every retrieved component. All referencing, renaming, and instantiation processes are encapsulated in this module according to the requirements of the prototyped system.

Third, a complete graphical user interface has been designed and implemented to demonstrate and manipulate the capabilities of the software base. All software base operations such as search, retrieval, and integration can be performed through the software base GUI.

There are still many areas that can be improved by future research. The following section describes those areas and provides recommendations that may help solve some of the problems.

B. FUTURE WORK

1. Providing Components For Software Base

The components to be added to the software base must be thoroughly tested to ensure that they meet their specifications. Testing of software components is still a

challenging area and requires further research. This is necessary in order to make reusable software libraries more successful.

The components that should be incorporated to the CAPS software base must be specified in PSDL. A study for the automation of this process can be found in [Seal92]. Application of this study to some of the existing software sources such as Booch library, RAPID project, and Ada software repository should provide plenty of good quality software components to be utilized by the software base.

2. Integrating Generic Components

Currently, the generic parameter values of an integrated component is supplied by user. This process can be automated by including an internal matching/mapping mechanism to the software base. One example of this process is the employment of OBJ3 specification in [Ste91]. Other mechanisms may be developed that perform the mapping of these parameters and matching them with the provided values for generic instantiation.

3. Ensuring Consistency

Updating and deleting components from the software base may cause system inconsistencies. These inconsistencies may result from the interdependencies of the components in the software base or from the dependency of the previously developed prototypes to the retrieved components from the software base. Currently, software base administrator manages the consistency of the system. This process can be automated by utilizing a version control schema.

4. Seperate GUI For User And Administrator

The software base GUI designed and implemented in this thesis demonstrates the full functionality of the CAPS software base. However, an ordinary user will not need such functions of software base as adding and deleting software libraries and components, updating existing components, etc. While this detailed software base interface is utilized by

system administrator, a simpler interface may be designed, emphasizing only those aspects of software base which will be frequently used by users.

LIST OF REFERENCES

- [Ada83] Department of Defence ANSI/MIL-STD-1815A, *Reference Manual for the Ada Programming Language*, Ada Joint Program Office, January 1983.
- [Bass87] Basset, P. G., *Unifying the Life-Cycle: CASEing Reusability*, CASE 87, Advance Papers, Vol 1, 1987.
- [Bass88] Basset, P. G., *Reusability in Software Design, Construction and Maintenance*, CASE 88, Advance Papers, Vol 2, 1988.
- [BL88] Berzins, V. and Luqi, *Rapidly Prototyping Real-Time Systems*, IEEE Software, September 1988.
- [BL91] Berzins, V. and Luqi, *Software Engineering with Abstractions*, Addison-Wesley, Reading, MA, 1991.
- [BLY88] Berzins, V., Luqi and Yeh, R., *A Prototyping Language for Real-Time Software*, IEEE Transactions on Software Engineering, Vol. 14, pp.1409, October 1988.
- [Booc86] Booch, Grady, *Software Components with Ada, Second Edition*, The Benjamin/Cummings Publishing Company, 1986.
- [Booc87] Booch, Grady, *Software Engineering with Ada, Structures, Tools and Subsystems*, The Benjamin/Cummings Publishing Company, 1987.
- [Booc90] Booch, Grady, *Library of Reusable Ada Components*, Wizard Software, Lakewood, CO, 1990.
- [BP89] Biggerstaff, T. J. and Perlis, A. J., *Software Reusability Volume I Concepts and Models*, Addison-Wesley, 1989.
- [BW87] Burton, Bruce A., Wienk, Rhonda, and others, *The Reusable Software Library*, IEEE Software, V. 4, pp. 25-33, July 1987.
- [Cox90] Cox, B. J., *Planning the Software Industrial Revolution*, IEEE Software, Sept. 1990.
- [FH87] Ferrans, J. C. and Hurst, D., *Facilitating Software Reuse via Automated Software Librarians*, CASE 87, Advance Papers, Vol 1, 1987.
- [GTW78] Goguen, J. A., Thatcher, J.W. and Wagner, E.G., *An Initial Approach to the Specification, Correctness, and Implementation of Abstract Data Types*, Current Trends in Programming Methodology, Volume IV, Raymond T. Yeh, ed, Prentice-Hall, Inc., 1978.

- [Koeg87] Koegel, J. F., *Application of Enabling Technologies to Software Reusability*, CASE 87, Advance Papers, Vol 1, 1987.
- [KL91] Katwijk, Jan van and Li, Haikuan, *The Steps Towards Reuse-in-the-Large*, Delft University Technology, Delft, Netherlands, 1991.
- [LK88] Luqi and Ketabchi, M., *A Computer Aided Prototyping System*, IEEE Software, March 1988.
- [Luba88] Lubars, M. D., *Environmental Support for Reuse*, CASE 88, Advance Papers, Vol 2, 1988.
- [Luqi88] Luqi, *Knowledge-Based Support for Rapid Prototyping*, IEEE Expert, Winter 1988.
- [Luqi89] Luqi, *Software Evolution via Rapid Prototyping*, IEEE Computer, May 1989.
- [Luqi91] Luqi, *Computer Aided Software Prototyping*, IEEE Computer, September 1991.
- [McDo91] McDowell, J. K., *A Reusable Component Retrieval System for Prototyping*, Master's Thesis, Naval Postgraduate School, September 1991.
- [Meye88] Meyer, Bertrand, *Object-Oriented Software Construction*, Prentice Hall, 1988.
- [Nasa91] NASA Goddard Space Flight Center, *Release Notes for TAE Plus*, ver. 5.1, FL, April 1991.
- [Neig84] Neighbors, James M., *The Draco Approach to Constructing Software from Reusable Components*, IEEE Transactions on Software Engineering, V.10, pp. 564-574, September 1984.
- [Onto90] Ontologic Inc., *Ontos Object Database Documentation Release 2.0 Beta*, Burlington, MA, October 1990.
- [RW90a] Rich, Charles, and Waters, Richard C., *The Programmer's Apprentice*, Addison-Wesley, 1990.
- [RW90b] Rollins, Eugene J., and Wing, Jeanette M., *Specifications as Search Keys for SW Libraries: A Case Study using Lambda Prolog*, CMU-CS-90-159, Carnegie Mellon University, 26 September 1990.
- [Seal92] Sealander, J., Master's Thesis, Naval Postgraduate School, September 1992.
- [Self90] Self, J., *Aflex - An Ada Lexical Analyzer Generator*, ver. 1.1, Arcadia Document UCI-90-18, University of California, Irvine, CA, May 1990.

- [SM83] Salton, G. and McGill, M., *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.
- [Ste91] Steigerwald, R. A., *Reusable Component Retrieval via Normalized Algebraic Specifications*, Ph.D. Dissertation, Naval Postgraduate School, December 1991.
- [TTS88] Taback, D., Tolani, T., and Schmalz, R.J., *Ayacc User's Manual*, ver. 1.0, Arcadia Document UCI-85-10, University of California, Irvine, CA, May 1988.
- [Voge90] Vogelsong, T., and Rothrock, J., *Reusable Ada Products for Information Systems Development (RAPID) Lessons Learned During Pilot Operations*, U.S. Army Information Systems Software Development Center, Washington, 1990.
- [YT88] Yin, W. P. and Tanik, M. M., *Reusability in the Real-Time Use of Ada*, CASE 88, Advance Papers, Vol 2, 1988.

APPENDIX A - INTEGRATION EXAMPLES

In this appendix various examples of integration of components are given. The component chosen as an example is the partial implementation of a set ADT. Two operators are included in it and two versions of the same component is provided : generic and non-generic.

A. INTEGRATING A COMPONENT RETRIEVED BY NAMED LOOKUP

5. Retrieving a Non-generic Component

a. PSDL Specification of Retrieved Component

```
type set
specification

operator empty
specification
  input  si:set
  output s:set
end

operator add
specification
  input  x:t, ,
         si:set
  output so:set
end

keywords set
description { set } end
implementation ada set_sb end
```

b. *Proto.set.a*

```
with set_sb;

package set_pkg is

    procedure empty ( si : in set;
                     s  : out set )
    renames set_sb.empty;

    procedure add ( x : in t;
                   si: in set;
                   so: out set )
    renames set_sb.add;

end set_pkg;
```

6. Retrieving a Generic Component

a. *PSDL Specification of Retrieved Component*

```
type set
specification
generic
    t:GENERIC_TYPE,

operator empty
specification
    output s:set
end

operator add
specification
    input  x:t,
           si:set
    output so:set
end

keywords set
description { set } end
implementation ada set_sb end
```

b. *Proto.set.a*

```
with set_sb;

package set_pkg is

    package tmp_set_pkg is new set_sb ( gen_p );

    procedure empty ( s : out set )
    renames tmp_set_pkg.empty;

    procedure add ( x : in gen_p;
                    si : in set;
                    so : out set )
    renames tmp_set_pkg.add;

end set_pkg;
```

B. INTEGRATING A COMPONENT RETRIEVED BY PSDL QUERY

1. Retrieving a Non-generic Component

a. *PSDL Specification of Retrieved Component*

```
type set
specification

operator empty
specification
    input  si:set
    output s:set
end

operator add
specification
    input  x:integer,
           si:set
    output so:set
end

keywords set
description { set } end
implementation ada set_sb end
```


b. PSDL Query

```
type integer_set
specification

    operator create
    specification
        input  in_set :set
        output out_set :set
    end

    operator put
    specification
        input  item  :integer,
              in_set :set
        output out_set:set
    end
end
implementation ada integer_set end
```

c. *Proto.integer_set.a*

```
with set_sb;

package integer_set_pkg is

    procedure create ( inset   : in set;
                     out_set  : out set )
    renames set_sb.empty;

    procedure put ( item   : in integer;
                  in_set  : in set;
                  out_set : out set )
    renames set_sb.add;

end integer_set_pkg;
```

2. Retrieving a Generic Component

a. PSDL Specification of Retrieved Component

```
type set
specification
generic
    t:GENERIC_TYPE,

operator empty
specification
    output s:set
end

operator add
specification
    input  x:t,
           si:set
    output so:set
end

keywords set
description { set } end
implementation ada set_sb end
```

b. PSDL Query

```
type int_set
specification

    operator create
    specification
        input  in_set :set
        output out_set :set
    end

    operator put
    specification
        input  item :integer,
               in_set :set
        output out_set:set
    end
end
implementation ada int_set end
```

c. *Proto.int_set.a*

```
with set_sb;

package int_set_pkg is

    package tmp_int_set_pkg is new set_sb(integer);

    procedure create ( inset    : in set;
                       out_set  : out set )
    renames tmp_int_set_pkg.empty;

    procedure put ( item    : in integer;
                   in_set   : in set;
                   out_set  : out set )
    renames tmp_int_set_pkg.add;

end int_set_pkg;
```

APPENDIX B - SOFTWARE BASE

GRAPHICAL USER INTERFACE USER'S MANUAL

The software base graphical user interface is implemented by using Transportable Applications Environment (TAE) Plus, an integrated environment for developing and running contemporary window and graphics based application systems.

The Software base GUI can be started by executing the command `softbase.exe`, by passing two command line parameters. First parameter is the path to the prototype directory and the second parameter is the prototype prefix, which is required by the software base script for compilation. Because of the required environment variables, it should be invoked while working on `sun51` or `rxterm`'ed to `sun51`. This requirement is due to the current Ontos libraries being on `sun51`.

The terminology used in this appendix is consistent with TAE Plus documentation.

A. LIBRARY FUNCTIONS

Library functions panel is the top level of the software base GUI system. It provides three kinds of functionality: *Library Operations*, *Quit*, and *Help*.

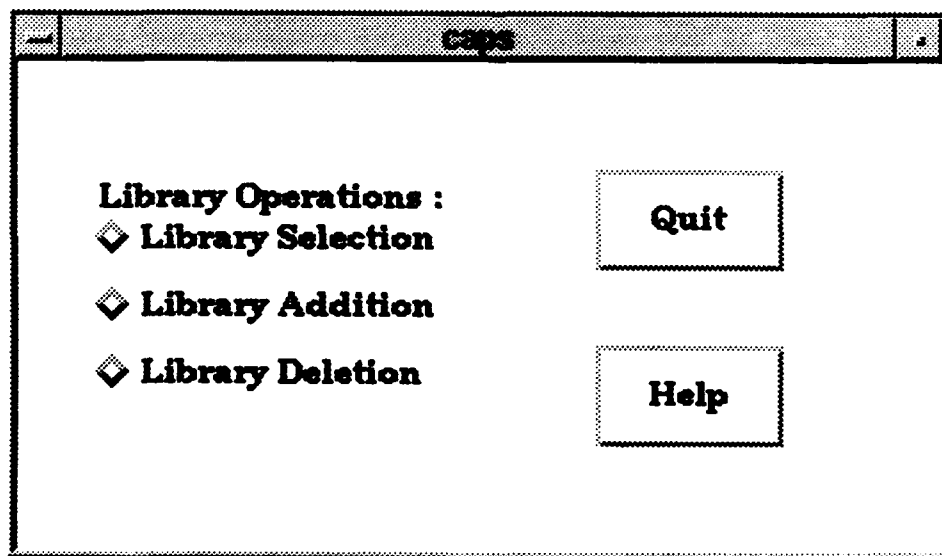


Figure B.1 - Library Functions Panel

Library Operations option provides three choices which are *Library Selection*, i.e. selecting the library that is going to be used throughout the software base manipulations (Ada1, Ada2, C++, etc.), *Library Addition*, in case creation of a new library is desired, and *Library Deletion*, if the deletion of an existing library from the software base is desired.

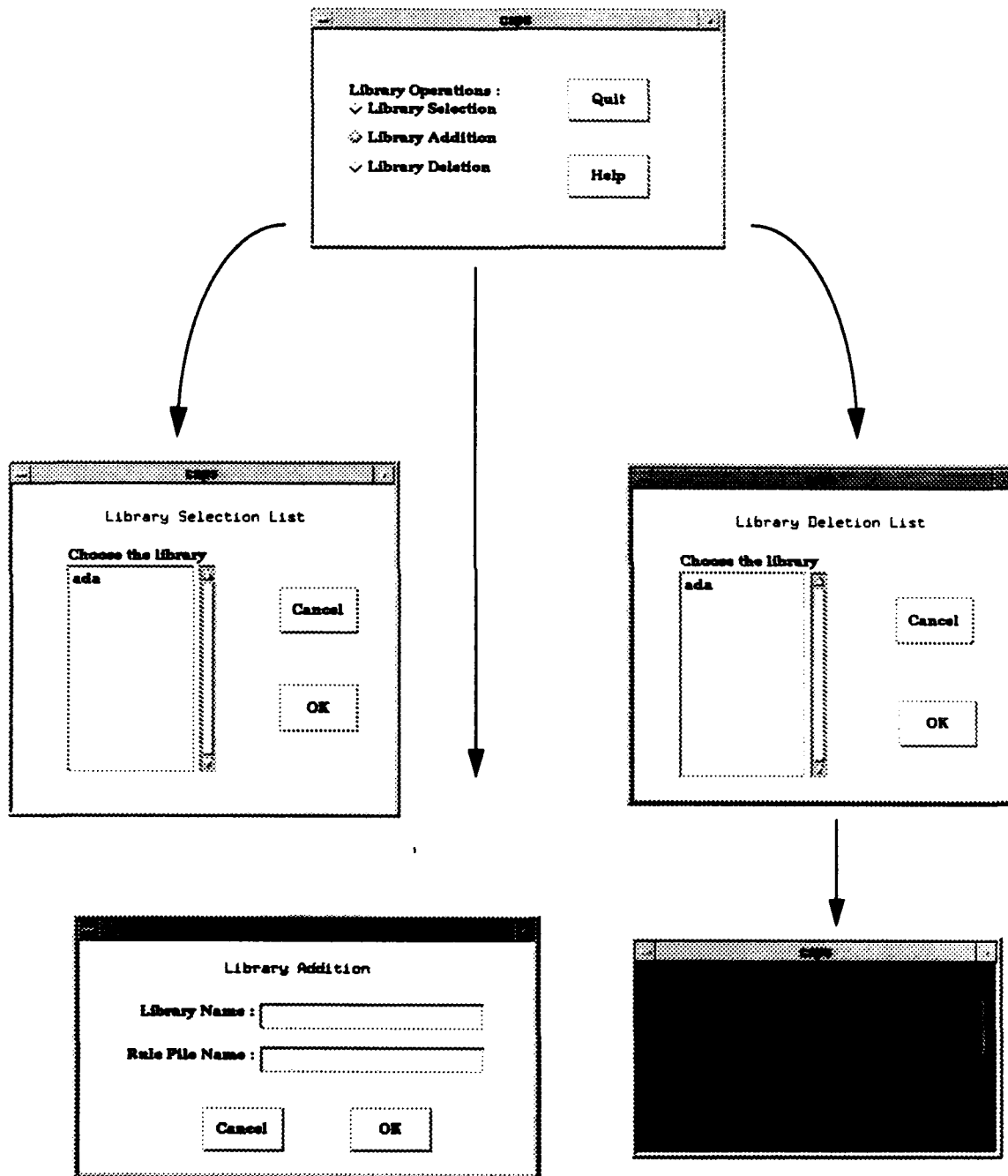


Figure B.2 - Library Functions Panel Connections

Library Selection panel displays the existing libraries in the software base. Clicking one of the libraries, for example Ada, results in all the operations to be performed on that particular library. It can be considered as a library domain selector. After clicking on the OK button, *Main Menu* panel is displayed to perform further software base operations.

Library Addition panel is provided to add new library domains into the software base. User keys in the new library name and the associated rule file name. Upon clicking on the OK button a new library is created and displayed in the library selection panel. This option is provided only for the authorized personnel (root).

Library Deletion panel is used to delete a library from the software base. This panel displays the existing library list. Selecting one of the libraries and clicking on OK button brings a warning message. Validating the operation deletes the selected library and all of the components in that library from the software base. This option is provided only for the authorized personnel (root).

Quit button terminates the software base GUI.

Clicking on *Help* button changes the mouse cursor into a '?' mark. Clicking the cursor on the desired item displays a help window about that item. For example, clicking on Library Operations brings a help window informing about the functionality of each operation.

B. MAIN MENU

Main Menu panel provides the basic operations of software base.

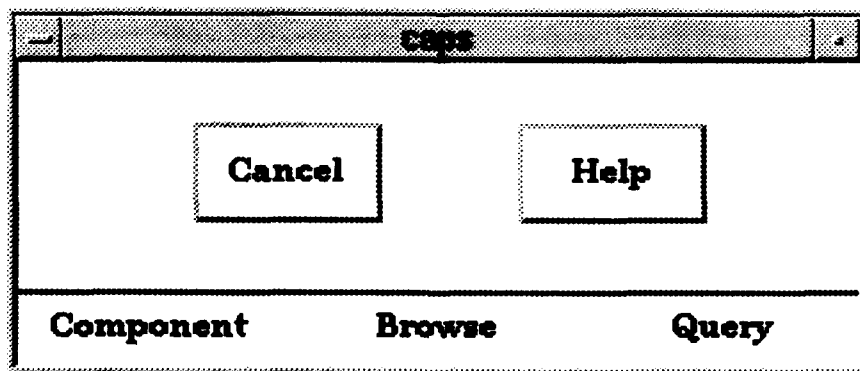


Figure B.3 - Main Menu Panel

These operations, implemented as pull down menu, include *Component Add/Update*, *Browse Types/Operators*, and *Keyword/PSDL Query*.

7. Component Add/Update

Clicking on *Component Add/Update* brings the *Input File Selection* panel.

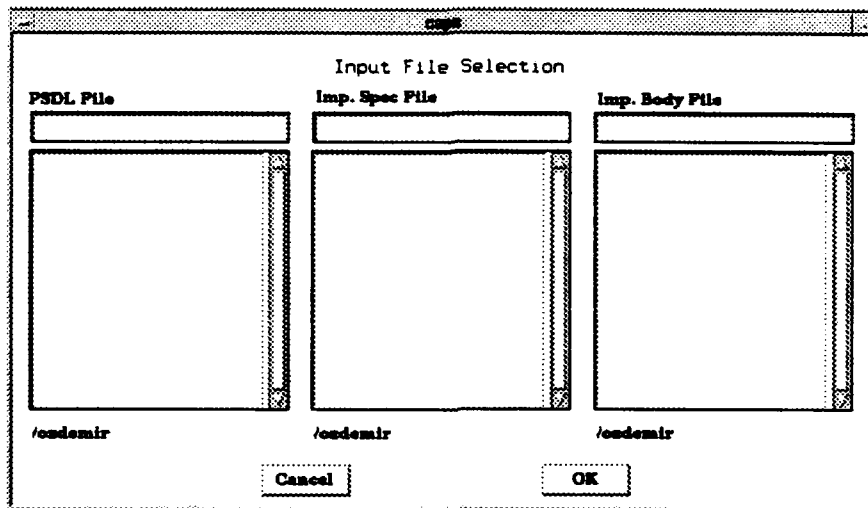


Figure B.4 - Input File Selection Panel

Inputs to the software base are made up of three text files. The PSDL specification, the implementation specification, and the implementation body. This panel allows the selection of each of these files.

The directory box displays the files and subdirectories in the current working directory. Under this box, the directory label displays the name of the current directory. Clicking on a directory in directory box will change to that directory and the path for the new directory will be displayed in the directory label. Clicking on a file will select that file and the selected file will be displayed in the file label box. After all the files are selected, clicking on OK button will cause the input files to be processed.

If *Component Add* operation is invoked, the system will attempt to add the component to the software base. In case of an error, an error message will be displayed.

Otherwise, the panel will be removed and *Main Menu* panel will be displayed indicating a successful addition.

If *Component Update* operation is invoked, the new PSDL specification and implementation files will replace those that are currently in the software base.

Cancel button will cause the input operation to terminate.

8. Browse Types/Operators

Clicking on *Browse Types/Operators* brings the *Component Selection* panel.

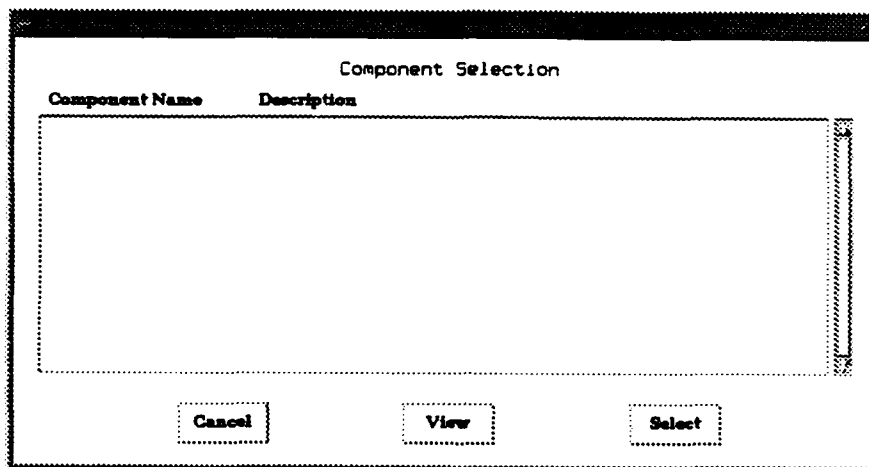


Figure B.5 - Component Selection Panel

Depending on the selection, this panel displays a list of all the operator or type components in the software base and a one line description of each component. These components are ordered alphabetically.

Selecting a component and clicking on *View* button will bring up a view of that components PSDL specification. See section B.C for details on using *View* panel.

Selecting a component and clicking on *Select* button will bring up a menu of more detailed operations on the selected component. See section B.D for details about this option.

Cancel button will cause the panel to disappear and brings up the *Main Menu* panel.

9. Keyword/PSDL Query

a. *Keyword Query*

Clicking on keyword query brings up the *Keyword Selection* panel.

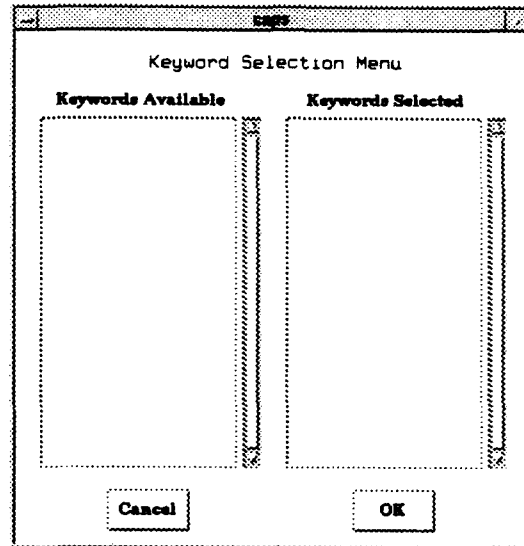


Figure B.6 - Keyword Selection panel

This panel allows the selection of keywords for a keyword search of the software base. Existing keywords in the software base are listed in the left box. Clicking on a keyword will add it to the *Keyword Selected* box. Once the desired keywords are selected, pushing the OK button will start the search. The resulting list of the components found in software base will be displayed in the *Component Selection* panel (See B.B.2).

Pushing the *cancel* button will terminate the keyword query.

b. *PSDL Query*

Clicking on PSDL query will bring up the *Query File Selection* panel.

Input to the software base is a PSDL specification file to conduct the query. This panel allows the selection of PSDL file. The directory box displays the files and subdirectories in the current working directory. Under this box, the directory label displays

the name of the current directory. Clicking on a directory in directory box will change to that directory and the path for the new directory will be displayed in the directory label. Clicking on a PSDL file will select that file and the selected file will be displayed in the file label box. After the query PSDL file is selected, clicking on OK button will cause the query to be processed. If any components were found that match the query specification then the *Component Selection* panel (See B.B.2) will be displayed providing a list of matching components. Otherwise, a message will be displayed stating that no component is found.

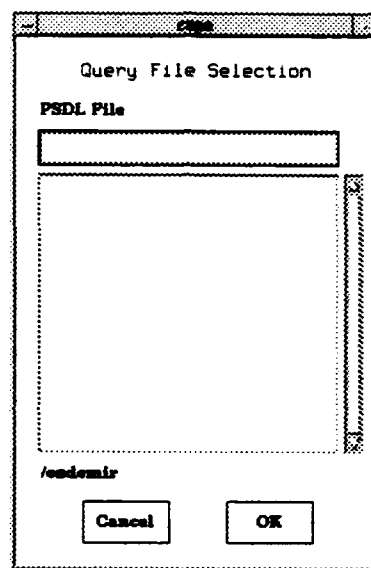


Figure B.7 - Query File Selection Panel

C. VIEW COMPONENT

After a component is selected from *Component Selection* panel, one of the operations that can be performed on the component is to view the files stored in the software base. These files are the PSDL specification, the implementation specification, and the implementation body files.

Clicking on *View* button on the *Component Selection* panel brings up the *PSDL Specification Viewing* panel. If no component is selected a warning message is displayed.

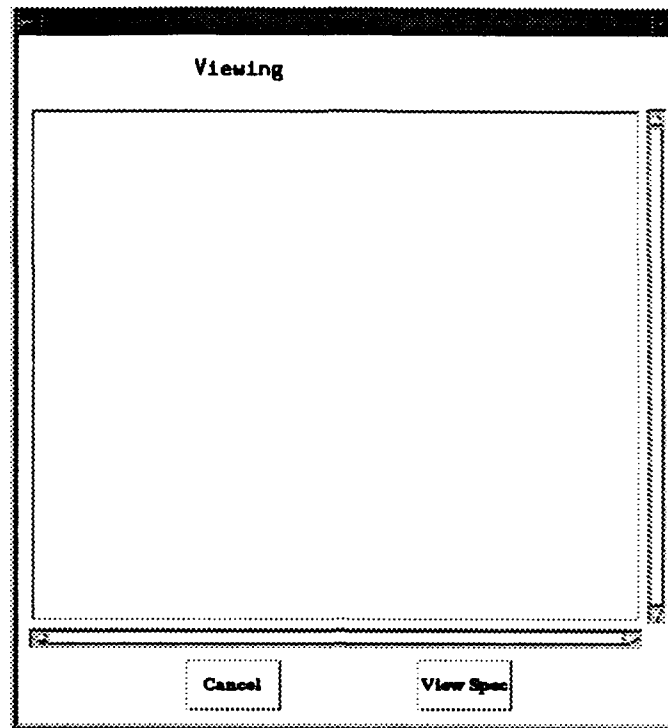


Figure B.8 - PSDL Specification Viewing Panel

This panel displays the PSDL specification for a given component. The name of the file displayed appears on the top of the display box.

By clicking on the *View Spec* button, implementation specification of the selected component can be viewed in a similar fashion, and consequently the implementation body file. Because the viewing panels are not removed, all three files can be viewed simultaneously. To remove the panels, *Cancel* buttons should be pushed.

As implementation specification and implementation body view panels are very similar to this panel, they are not included in the figures.

D. SELECT COMPONENT

Clicking on *Select* button on the *Component Selection* panel brings up the *Select* panel. If no component is selected a warning message is displayed.

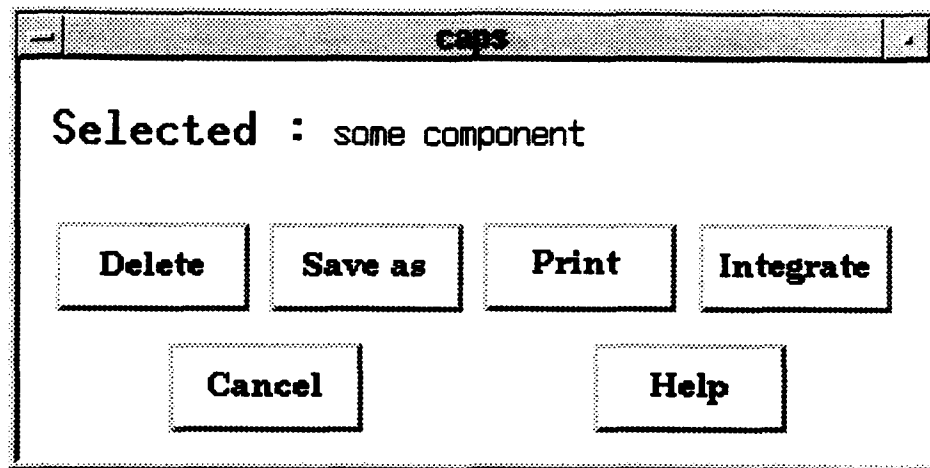


Figure B.9 - Select Panel

This panel enables user to perform various operations on the selected component.

Clicking on *Delete* button causes the component to be deleted from the software base after displaying a warning message.

Clicking on *Save as* button brings up the *Save Component* panel.

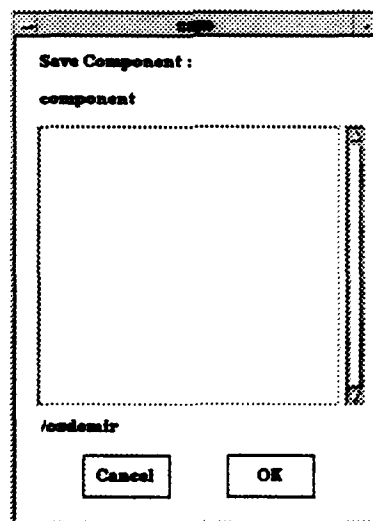


Figure B.10 - Save Component Panel

This panel enables the user to save the selected component in any workplace. All three files, PSDL specification, implementation specification and implementation body files, are

saved into the specified directory. Default is the prototype directory and path to this directory is obtained from the first argument of user interface executable.

Clicking on *Print* button displays three options: printing the PSDL file, printing the implementation specification, or printing the implementation body. Selected files are printed through the default printer.

Clicking on *Cancel* button terminates the panel.

Clicking on *Help* button changes the mouse cursor into a '?' mark. Clicking the cursor on the desired item displays a help window about that item.

Integrate button is used to incorporate the selected component into the system. Pushing this button brings up the *Integration* panel. The integrate button of this panel causes the integration tool to be enabled and resulting integration module (the package the code is encapsulated) is saved into the prototyping directory together with the retrieved component. Depending on the retrieved component being generic or non-generic, an information message is displayed in the box.

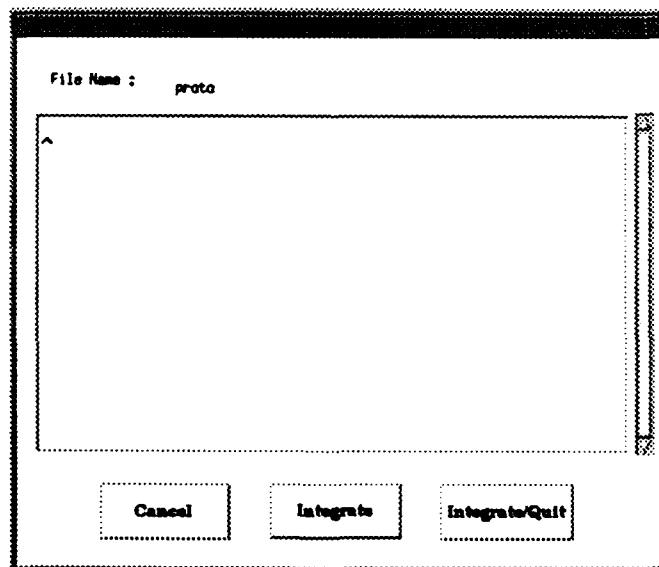


Figure B.11 - Integration Panel

The only difference between *Integrate* and *Integrate/Quit* button is while first button returns back to the *Main Menu* panel, the latter quits the program after integration process.

APPENDIX C - SOURCE CODE FOR INTEGRATION TOOL

A. PSDL_LEX.L

```
-- *** File      : psdl_lex.l
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

%START IDENT Z
Digit  [0-9]
Int    (Digit)+
Letter [a-zA-Z_]
Alpha  ((Letter)|(Digit))
Blank  [ \t\n]
Text   [^()]
StrLit  [^"\\]|["\\]["\\]
Quote  ["]
A       [aA]
B       [bB]
C       [cC]
D       [dD]
E       [eE]
F       [fF]
G       [gG]
H       [hH]
I       [iI]
J       [jJ]
K       [kK]
L       [lL]
M       [mM]
N       [nN]
O       [oO]
P       [pP]
Q       [qQ]
R       [rR]
S       [sS]
T       [tT]
U       [uU]
V       [vV]
W       [wW]
X       [xX]
Y       [yY]
Z       [zZ]
%%
{A}{D}{A}                (ENTER(Z);return(ADA_TOKEN); )
{A}{X}{I}{O}{M}{S}       (ENTER(Z);return(AXIOMS_TOKEN); )
{B}{Y}{Blank}+(A){L}{L}  (ENTER(Z);return(BY_ALL_TOKEN); )
```

```

{B}{Y}{Blank}+(S){O}{M}{E} {ENTER(Z);return(BY_SOME_TOKEN);}
{C}{O}{N}{T}{R}{O}{L} {ENTER(Z);return(CONTROL_TOKEN); }
{C}{O}{N}{S}{T}{R}{A}{I}{N}{T}{S} {ENTER(Z);return(CONSTRAINTS_TOKEN); }
{D}{A}{T}{A} {ENTER(Z);return(DATA_TOKEN); }
{D}{E}{S}{C}{R}{I}{P}{T}{I}{O}{N} {ENTER(Z);return(DESCRIPTION_TOKEN); }
{E}{D}{G}{E} {ENTER(Z);return(EDGE_TOKEN); }
{E}{N}{D} {ENTER(Z);return(END_TOKEN); }
{E}{X}{C}{E}{P}{T}{I}{O}{N}{S} {ENTER(Z);return(EXCEPTIONS_TOKEN); }
{E}{X}{C}{E}{P}{T}{I}{O}{N} {ENTER(Z);return(EXCEPTION_TOKEN); }
{F}{I}{N}{I}{S}{H} {ENTER(Z);return(FINISH_TOKEN); }
{W}{I}{T}{H}{I}{N} {ENTER(Z);return(WITHIN_TOKEN); }
{G}{E}{N}{E}{R}{I}{C} {ENTER(Z);return(GENERIC_TOKEN); }
{G}{R}{A}{P}{H} {ENTER(Z);return(GRAPH_TOKEN); }
{H}{O}{U}{R}{S} {ENTER(Z);return(HOURS_TOKEN); }
{I}{F} {ENTER(Z);return(IF_TOKEN); }
{I}{M}{P}{L}{E}{M}{E}{N}{T}{A}{T}{I}{O}{N} {ENTER(Z);return(IMPLEMENTATION_TOKEN); }
{I}{N}{I}{T}{I}{A}{L}{L}{Y} {ENTER(Z);return(INITIALY_TOKEN); }
{I}{N}{P}{U}{T} {ENTER(Z);return(INPUT_TOKEN); }
{K}{E}{Y}{W}{O}{R}{D}{S} {ENTER(Z);return(KEYWORDS_TOKEN); }
{M}{A}{X}{I}{M}{U}{M} {ENTER(Z);return(MAXIMUM_TOKEN); }
{E}{X}{E}{C}{U}{T}{I}{O}{N} {ENTER(Z);return(EXECUTION_TOKEN); }
{T}{I}{M}{E} {ENTER(Z);return(TIME_TOKEN); }
{R}{E}{S}{P}{O}{N}{S}{E} {ENTER(Z);return(RESPONSE_TOKEN); }
{M}{I}{C}{R}{O}{S}{E}{C} {ENTER(Z);return(MICROSEC_TOKEN); }
{M}{I}{N}{I}{M}{U}{M} {ENTER(Z);return(MINIMUM_TOKEN); }
{C}{A}{L}{L}{P}{E}{R}{I}{O}{D} {ENTER(Z);return(CALL_PERIOD_TOKEN); }
{M}{I}{N}{I}{M}{U}{M} {ENTER(Z);return(MIN_TOKEN); }
{M}{S} {ENTER(Z);return(MS_TOKEN); }
{O}{P}{E}{R}{A}{T}{O}{R} {ENTER(Z);return(OPERATOR_TOKEN); }
{O}{U}{T}{P}{U}{T} {ENTER(Z);return(OUTPUT_TOKEN); }
{P}{E}{R}{I}{O}{D} {ENTER(Z);return(PERIOD_TOKEN); }
{R}{E}{Q}{U}{I}{R}{E}{D}{B}{Y} {ENTER(Z);return(REQ_BY_TOKEN); }
{R}{E}{S}{E}{T} {ENTER(Z);return(RESET_TOKEN); }
{S}{E}{C}{O}{N}{D} {ENTER(Z);return(SEC_TOKEN); }
{S}{P}{E}{C}{I}{F}{I}{C}{A}{T}{I}{O}{N} {ENTER(Z);return(SPECIFICATION_TOKEN); }
{S}{T}{A}{R}{T} {ENTER(Z);return(START_TOKEN); }
{S}{T}{A}{T}{E}{S} {ENTER(Z);return(STATES_TOKEN); }
{S}{T}{O}{P} {ENTER(Z);return(STOP_TOKEN); }
{S}{T}{R}{E}{A}{M} {ENTER(Z);return(STREAM_TOKEN); }
{T}{I}{M}{E}{R} {ENTER(Z);return(TIMER_TOKEN); }
{T}{R}{I}{G}{G}{E}{R}{E}{D} {ENTER(Z);return(TRIGGERED_TOKEN); }
{T}{Y}{P}{E} {ENTER(Z);return(TYPE_TOKEN); }
{V}{E}{R}{T}{E}{X} {ENTER(Z);return(VERTEX_TOKEN); }
"and"|"AND" {ENTER(Z);return(AND_TOKEN); }
"or"|"OR" {ENTER(Z);return(OR_TOKEN); }
"xor"|"XOR" {ENTER(Z);return(XOR_TOKEN); }
">=" {ENTER(Z);return(GREATER_THAN_OR_EQUAL); }

```

```

"<="      {ENTER(Z);return(LESS_THAN_OR_EQUAL); }
"/="|"!=" {ENTER(Z);return(INEQUALITY); }
">"      {ENTER(Z);return(ARROW); }
"="      {ENTER(Z);return('='); }
"+"      {ENTER(Z);return('+'); }
"-"      {ENTER(Z);return('-'); }
"*"      {ENTER(Z);return('*'); }
"/"      {ENTER(Z);return('/'); }
"&"      {ENTER(Z);return('&'); }
"("      {ENTER(Z);return('('); }
")"      {ENTER(Z);return(')'); }
"["      {ENTER(Z);return('['); }
"]"      {ENTER(Z);return(']'); }
":"      {ENTER(Z);return(':'); }
","      {ENTER(Z);return(','); }
"."      {ENTER(Z);return('.'); }
"|"      {ENTER(Z);return('|'); }
">"      {ENTER(Z);return('>'); }
"<"      {ENTER(Z);return('<'); }
"mod"|"MOD" {ENTER(Z);return(MOD_TOKEN); }
"rem"|"REM" {ENTER(Z);return(REM_TOKEN); }
"***"|"EXP"|"exp" {ENTER(Z);return(EXP_TOKEN); }
"abs"|"ABS" {ENTER(Z);return(ABS_TOKEN); }
"not"|"NOT" {ENTER(Z);return(NOT_TOKEN); }
(T)(R)(U)(E) {ENTER(Z);return(TRUE); }
(F)(A)(L)(S)(E) {ENTER(Z);return(FALSE); }
<IDENT>\' {ENTER(Z);return(''); }
(Letter)(Alpha)* {ENTER(IDENT);

```

```

    strlen(psdl_lex_dfa.yytext,length);

```

```

    the_id_token(1..length) := psdl_lex_dfa.yytext;

```

```

    return(IDENTIFIER); }
(Quote){StrLit}*(Quote) {ENTER(Z);return(STRING_LITERAL); }
(Int) {ENTER(Z);return(INTEGER_LITERAL); }
(Int)"."(Int) {ENTER(Z);return(REAL_LITERAL); }
"({Text}*)" {ENTER(Z);return(TEXT_TOKEN); }
[\n] {NULL;}
[ \t] {NULL;} --ignore spaces and tabs
%%

```

```

with Psdl_Tokens,component_aux_pkg;
use Psdl_Tokens,component_aux_pkg;

```

```

package Psdl_Lex is
    function yylex return token;
end Psdl_Lex;

```

```

package body Psdl_Lex is
    ##
end Psdl_Lex;

```


B. PSDL.Y

```
-- *** File      : psdl.y
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

%token '(' ')' '[' ']' ':' ',' ' ' '.' '|'
%token ARROW
%token TRUE FALSE
%token ADA_TOKEN AXIOMS_TOKEN
%token BY_ALL_TOKEN REQ_BY_TOKEN BY_SOME_TOKEN
%token CALL_PERIOD_TOKEN CONTROL_TOKEN
%token CONSTRAINTS_TOKEN
%token DESCRIPTION_TOKEN DATA_TOKEN
%token END_TOKEN EDGE_TOKEN EXCEPTION_TOKEN
%token EXCEPTIONS_TOKEN EXECUTION_TOKEN
%token FINISH_TOKEN
%token GENERIC_TOKEN GRAPH_TOKEN
%token HOURS_TOKEN
%token IF_TOKEN IMPLEMENTATION_TOKEN
%token INPUT_TOKEN INITIALLY_TOKEN
%token KEYWORDS_TOKEN
%token MAXIMUM_TOKEN MINIMUM_TOKEN
%token MICROSEC_TOKEN MIN_TOKEN MS_TOKEN
%token MOD_TOKEN NOT_TOKEN
%token OPERATOR_TOKEN OR_TOKEN
%token OUTPUT_TOKEN
%token PERIOD_TOKEN
%token RESET_TOKEN RESPONSE_TOKEN
%token SPECIFICATION_TOKEN SEC_TOKEN
%token START_TOKEN STATES_TOKEN STOP_TOKEN
%token STREAM_TOKEN
%token TIME_TOKEN TIMER_TOKEN
%token TYPE_TOKEN TRIGGERED_TOKEN
%token VERTEX_TOKEN
%token WITHIN_TOKEN
%token TEXT_TOKEN
%token IDENTIFIER
%token STRING_LITERAL
%token INTEGER_LITERAL
%token REAL_LITERAL

--operator precedences
%left AND_TOKEN OR_TOKEN XOR_TOKEN LOGICAL_OPERATOR
%left '<' '>' '=' GREATER_THAN_OR_EQUAL LESS_THAN_OR_EQUAL INEQUALITY
RELATIONAL_OPERATOR
%left '+' '-' '&' BINARY_ADDING_OPERATOR
%left UNARY_ADDING_OPERATOR
%left '*' '/' MOD_TOKEN REM_TOKEN MULTIPLYING_OPERATOR
%left EXP_TOKEN ABS_TOKEN NOT_TOKEN HIGHEST_PRECEDENCE_OPERATOR
```

```

%start start_symbol
(
  subtype yystype is integer;
)

%%
start_symbol:
    psdl
    ;
psdl:
    psdl
    component
    |
    ;
component:
    data_type
    |
    operator
    ;
data_type:
    TYPE_TOKEN
    (
    is_adt :=True;
    )
    IDENTIFIER
    (
        null;
    )
    type_spec type_impl
    ;
type_spec:
    SPECIFICATION_TOKEN optional_generic_param
optional_type_decl
    op_spec_list functionality END_TOKEN
    ;
optional_generic_param:
    GENERIC_TOKEN ,
    (
        is_generic := True;
        gen_par := True;
    )
    list_of_type_decl
    |
    ;
optional_type_decl:
    list_of_type_decl
    |
    ;
op_spec_list:
    op_spec_list
    OPERATOR_TOKEN

```

```

IDENTIFIER
(
    tmp_op_name(1..length):=the_id_token(1..length);
len_name:=length;
)
operator_spec
(
    if is_adt then
        component_aux_pkg.Put_adt_op
(tmp_op_name, len_name, has_inputs, has_outputs, adt_op_ptr);
        has_inputs:= False;
        has_outputs:= False;
        in_par:=False;
        out_par:=False;
        end if;
    )
;
operator:
OPERATOR_TOKEN
(
    is_operator := True;
)
IDENTIFIER
(
    null;
)
operator_spec operator_impl
;
operator_spec:
SPECIFICATION_TOKEN
interface
functionality
END_TOKEN
;
interface:
interface attribute reqmts_trace
|
;
attribute:
GENERIC_TOKEN
(
    is_generic := True;
    gen_par := True;
)
list_of_type_decl
|
INPUT_TOKEN
(
    has_in_param := True;

```

```

has_inputs    := True;
in_par        := True;
gen_par       := False;
}
list_of_type_decl
|
OUTPUT_TOKEN
(
    has_out_param := True;
    has_outputs   := True;
    in_par        := False;
    out_par       := True;
    gen_par       := False;
)
list_of_type_decl
|
STATES_TOKEN
list_of_type_decl
INITIALLY_TOKEN
initial_expression_list
|
EXCEPTIONS_TOKEN
id_list
|
MAXIMUM_TOKEN EXECUTION_TOKEN TIME_TOKEN time
;
list_of_type_decl:
    list_of_type_decl ',' type_decl
    |
    type_decl
    ;
type_decl:
    id_list
    (
        if gen_par then
            if the_id_token(1..length) = "BASE_TYPE" or
               the_id_token(1..length) = "base_type" then
                null;
            else
                component_aux_pkg.Put_gen_parameter(the_id_token,
                                                       " ", length, 1, gen_par_ptr);
            end if;
        end if;

        if in_par then
            if out_par then
                null;
            else
                component_aux_pkg.Put_in_parameter(the_id_token, length, in_par_ptr);
            end if;
        end if;
    )

```

```

        if out_par then
component_aux_pkg.Put_out_parameter(the_id_token,length,out_par_ptr);
        end if;
    )
        ':'
type_name
    (
        if in_par then
            if out_par then
                null;
            else
                component_aux_pkg.Put_in_parameter_val(the_id_token,
                                                            length,in_par_val_ptr);
            end if;
        end if;

        if out_par then
            component_aux_pkg.Put_out_parameter_val(the_id_token,
                                                        length,out_par_val_ptr);
        end if;
    )
;

type_name:
    IDENTIFIER
    '['
list_of_type_decl
    ']'
    |
    IDENTIFIER
;

id_list:
    id_list
    ','
    IDENTIFIER
    |
    IDENTIFIER
;

reqmts_trace:
    REQ_BY_TOKEN id_list
    |
;

functionality:
    keywords informal_desc formal_desc
;

keywords:
    KEYWORDS_TOKEN
    (
        out_par := False;
        gen_par := False;--in case no in/out parameters
    null;

```

```

        )
        id_list
        |
        ;
informal_desc:
        DESCRIPTION_TOKEN TEXT_TOKEN
        |
        ;
formal_desc:
        AXIOMS_TOKEN TEXT_TOKEN
        |
        ;
type_impl:
        IMPLEMENTATION_TOKEN ADA_TOKEN IDENTIFIER
        (
        withname.name(1..length):=the_id_token(1..length);
        withname.len:=length;
        )
        END_TOKEN
        |
        IMPLEMENTATION_TOKEN type_name op_impl_list END_TOKEN
        ;
op_impl_list:
        op_impl_list OPERATOR_TOKEN IDENTIFIER operator_impl
        |
        ;
operator_impl:
        IMPLEMENTATION_TOKEN ADA_TOKEN IDENTIFIER
        (
        withname.name(1..length):=the_id_token(1..length);
        withname.len:=length;
        )
        END_TOKEN
        |
        IMPLEMENTATION_TOKEN psdl_impl END_TOKEN
        ;
psdl_impl:
        data_flow_diagram streams timers control_constraints
informal_desc
        ;
data_flow_diagram:
        GRAPH_TOKEN vertex_list edge_list
        ;
vertex_list:
        vertex_list VERTEX_TOKEN op_id optional_time
        |
        ;
edge_list:
        edge_list EDGE_TOKEN IDENTIFIER
        optional_time op_id ARROW op_id
        |

```

```

op_id:
    ;
    IDENTIFIER opt_arg
    ;
opt_arg:
    '(' optional_id_list
    '|' optional_id_list ')'
    |
    ;
optional_id_list:
    id_list
    |
    ;
optional_time:
    ':' time
    |
    ;
streams:
    DATA_TOKEN STREAM_TOKEN list_of_type_decl
    |
    ;
timers:
    TIMER_TOKEN id_list
    |
    ;
control_constraints:
    CONTROL_TOKEN CONSTRAINTS_TOKEN constraints
    ;
constraints:
    constraints OPERATOR_TOKEN IDENTIFIER
    opt_trigger opt_period opt_finish_within
    opt_mcp opt_mrt constraint_options
    |
    OPERATOR_TOKEN IDENTIFIER
    opt_trigger opt_period opt_finish_within
    opt_mcp opt_mrt
    ;
constraint_options:
    constraint_options OUTPUT_TOKEN id_list IF_TOKEN
    expression reqmts_trace
    |
    constraint_options EXCEPTION_TOKEN IDENTIFIER
    opt_if_predicate reqmts_trace
    |
    constraint_options timer_op IDENTIFIER
    opt_if_predicate reqmts_trace
    |
    ;
opt_trigger:
    TRIGGERED_TOKEN trigger opt_if_predicate reqmts_trace
    |

```

```

trigger:
    ;
    BY_ALL_TOKEN id_list
    |
    BY_SOME_TOKEN id_list
    ;
opt_period:
    PERIOD_TOKEN time reqmts_trace
    ;
opt_finish_within:
    FINISH_TOKEN WITHIN_TOKEN time reqmts_trace
    ;
opt_mcp:
    MINIMUM_TOKEN CALL_PERIOD_TOKEN time reqmts_trace
    ;
opt_mrt:
    max_resp_time time reqmts_trace
    ;
max_resp_time:
    MAXIMUM_TOKEN RESPONSE_TOKEN TIME_TOKEN
    ;
timer_op:
    RESET_TOKEN
    |
    START_TOKEN
    |
    STOP_TOKEN
    ;
opt_if_predicate:
    IF_TOKEN expression
    ;
initial_expression_list:
    initial_expression_list ',' initial_expression
    |
    initial_expression
    ;
initial_expression:
    TRUE
    |
    FALSE
    |
    INTEGER_LITERAL
    |
    REAL_LITERAL
    ;

```



```

        STRING_LITERAL
        |
        IDENTIFIER
        |
        type_name '.' IDENTIFIER
        |
        type_name '.' IDENTIFIER '(' initial_expression_list ')'
        |
        '(' initial_expression_list ')'
        |
        initial_expression log_op initial_expression %prec
logical_operator
        |
        initial_expression rel_op initial_expression %prec
relational_operator
        |
        '-' initial_expression %prec unary_adding_operator
        |
        '+' initial_expression %prec unary_adding_operator
        |
        initial_expression bin_add_op initial_expression %prec
multiplying_operator
        |
        initial_expression bin_mul_op initial_expression %prec
multiplying_operator
        |
        initial_expression EXP_TOKEN initial_expression %prec
highest_precedence_operator
        |
        NOT_TOKEN initial_expression %prec
highest_precedence_operator
        |
        ABS_TOKEN initial_expression %prec
highest_precedence_operator
        ;
log_op:
        AND_TOKEN
        |
        OR_TOKEN
        |
        XOR_TOKEN
        ;
rel_op:
        '<' | '>' | '=' | GREATER_THAN_OR_EQUAL
        |
        LESS_THAN_OR_EQUAL | INEQUALITY
        ;
bin_add_op:
        '+' | '-' | '&'
        ;
bin_mul_op:

```

```

        '*' | '/' | MOD_TOKEN | REM_TOKEN
    ;

time:
    time_number MICROSEC_TOKEN
    |
    time_number MS_TOKEN
    |
    time_number SEC_TOKEN
    |
    time_number MIN_TOKEN
    |
    time_number HOURS_TOKEN
    ;

time_number:
    INTEGER_LITERAL
    ;

expression_list:
    expression_list ',' expression
    |
    expression
    ;

expression:
    TRUE | FALSE | INTEGER_LITERAL | time | REAL_LITERAL
    | STRING_LITERAL | IDENTIFIER
    | type_name '.' IDENTIFIER
    | type_name '.' IDENTIFIER '(' expression_list ')'
    | '(' expression_list ')'
    | expression log_op expression %prec logical_operator
    | expression rel_op expression %prec relational_operator
    | '-' expression %prec unary_adding_operator
    | '+' expression %prec unary_adding_operator
    | expression bin_add_op expression %prec
unary_adding_operator
    | expression bin_mul_op expression %prec
multiplying_operator
    | expression EXP_TOKEN expression %prec
highest_precedence_operator
    | NOT_TOKEN expression %prec highest_precedence_operator
    | ABS_TOKEN expression %prec highest_precedence_operator
    ;

%%

with Text_io,psdl_tokens,psdl_goto,psdl_shift_reduce,psdl_lex,
    component_aux_pkg;
use Text_io,psdl_tokens,psdl_goto,psdl_shift_reduce,psdl_lex,
    component_aux_pkg;

package component_parser is

    procedure Yyparse;

```

```

echo          : boolean := false;
number_of_errors : Natural := 0;
in_par        : Boolean:=False;
out_par       : Boolean:=False;
gen_par       : Boolean:=False;
has_inputs    : Boolean:=False;
has_outputs   : Boolean:=False;
tmp_op_name   : String(1..80);
len_name      : Integer:=0;

end component_parser;

package body component_parser is

  procedure Yerror(s : in string := "syntax error") is
  begin
    number_of_errors := number_of_errors + 1;
    put ("*** ");
    put_line(s);
  end Yerror;

  ##%procedure_parse

end component_parser;

```

C. QUERY_LEX.L

```
-- *** File      : query_lex.l
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

%START IDENT Z
Digit    [0-9]
Int      (Digit)+
Letter   [a-zA-Z_]
Alpha    ({Letter}|(Digit))
Blank    [ \t\n]
Text     [^()]
StrLit   [^"\\]|["\\]["\\]
Quote    ["]
A        [aA]
B        [bB]
C        [cC]
D        [dD]
E        [eE]
F        [fF]
G        [gG]
H        [hH]
I        [iI]
J        [jJ]
K        [kK]
L        [lL]
M        [mM]
N        [nN]
O        [oO]
P        [pP]
Q        [qQ]
R        [rR]
S        [sS]
T        [tT]
U        [uU]
V        [vV]
W        [wW]
X        [xX]
Y        [yY]
Z        [zZ]

%%

(A){D}(A)                (ENTER(Z);return(ADA_TOKEN); )
(A){X}(I){O}(M){S}       (ENTER(Z);return(AXIOMS_TOKEN); )
(B){Y}{Blank}+(A){L}{L}  (ENTER(Z);return(BY_ALL_TOKEN); )
(B){Y}{Blank}+(S){O}(M){E} (ENTER(Z);return(BY_SOME_TOKEN); )
(C){O}(N){T}{R}{O}{L}    (ENTER(Z);return(CONTROL_TOKEN); )
(C){O}(N){S}{T}{R}{A}{I}{N}{T}{S} (ENTER(Z);return(CONSTRAINTS_TOKEN); )
```

```

(D){A}{T}{A} (ENTER(Z);return(DATA_TOKEN); )
(D){E}{S}{C}{R}{I}{P}{T}{I}{O}{N} (ENTER(Z);return(DESCRIPTION_TOKEN); )
(E){D}{G}{E} (ENTER(Z);return(EDGE_TOKEN); )
(E){N}{D} (ENTER(Z);return(END_TOKEN); )
(E){X}{C}{E}{P}{T}{I}{O}{N}{S} (ENTER(Z);return(EXCEPTIONS_TOKEN); )
(E){X}{C}{E}{P}{T}{I}{O}{N} (ENTER(Z);return(EXCEPTION_TOKEN); )
(F){I}{N}{I}{S}{H} (ENTER(Z);return(FINISH_TOKEN); )
(W){I}{T}{H}{I}{N} (ENTER(Z);return(WITHIN_TOKEN); )
(G){E}{N}{E}{R}{I}{C} (ENTER(Z);return(GENERIC_TOKEN); )
(G){R}{A}{P}{H} (ENTER(Z);return(GRAPH_TOKEN); )
(H){O}{U}{R}{S} (ENTER(Z);return(HOURS_TOKEN); )
(I){F} (ENTER(Z);return(IF_TOKEN); )
(I){M}{P}{L}{E}{M}{E}{N}{T}{A}{T}{I}{O}{N}
(ENTER(Z);return(IMPLEMENTATION_TOKEN); )
(I){N}{I}{T}{I}{A}{L}{L}{Y} (ENTER(Z);return(INITIALY_TOKEN); )
(I){N}{P}{U}{T} (ENTER(Z);return(INPUT_TOKEN); )
(K){E}{Y}{W}{O}{R}{D}{S} (ENTER(Z);return(KEYWORDS_TOKEN); )
(M){A}{X}{I}{M}{U}{M} (ENTER(Z);return(MAXIMUM_TOKEN); )
(E){X}{E}{C}{U}{T}{I}{O}{N} (ENTER(Z);return(EXECUTION_TOKEN); )
(T){I}{M}{E} (ENTER(Z);return(TIME_TOKEN); )
(R){E}{S}{P}{O}{N}{S}{E} (ENTER(Z);return(RESPONSE_TOKEN); )
(M){I}{C}{R}{O}{S}{E}{C}{I}{M}{I}{C}{R}{O}{S}{E}{C}{O}{N}{D}{S}
(ENTER(Z);return(MICROSEC_TOKEN); )
(M){I}{N}{I}{M}{U}{M} (ENTER(Z);return(MINIMUM_TOKEN); )
(C){A}{L}{L}{I}{N}{G}{(Blank)+(P){E}{R}{I}{O}{D}
(ENTER(Z);return(CALL_PERIOD_TOKEN); )
(M){I}{N}{I}{M}{I}{N}{U}{T}{E}{S}
(ENTER(Z);return(MIN_TOKEN); )
(M){S}{I}{L}{L}{I}{S}{E}{C}{O}{N}{D}{S} (ENTER(Z);return(MS_TOKEN);
)
(O){P}{E}{R}{A}{T}{O}{R} (ENTER(Z);return(OPERATOR_TOKEN); )
(O){U}{T}{P}{U}{T} (ENTER(Z);return(OUTPUT_TOKEN); )
(P){E}{R}{I}{O}{D} (ENTER(Z);return(PERIOD_TOKEN); )
(R){E}{Q}{U}{I}{R}{E}{D}{(Blank)+(B){Y} (ENTER(Z);return(REQ_BY_TOKEN); )
(R){E}{S}{E}{T}{(Blank)+(T){I}{M}{E}{R} (ENTER(Z);return(RESET_TOKEN); )
(S){E}{C}{I}{S}{E}{C}{O}{N}{D}{S} (ENTER(Z);return(SEC_TOKEN); )
(S){P}{E}{C}{I}{F}{I}{C}{A}{T}{I}{O}{N}
(ENTER(Z);return(SPECIFICATION_TOKEN); )
(S){T}{A}{R}{T}{(Blank)+(T){I}{M}{E}{R} (ENTER(Z);return(START_TOKEN); )
(S){T}{A}{T}{E}{S} (ENTER(Z);return(STATES_TOKEN); )
(S){T}{O}{P}{(Blank)+(T){I}{M}{E}{R} (ENTER(Z);return(STOP_TOKEN); )
(S){T}{R}{E}{A}{M} (ENTER(Z);return(STREAM_TOKEN); )
(T){I}{M}{E}{R} (ENTER(Z);return(TIMER_TOKEN); )
(T){R}{I}{G}{G}{E}{R}{E}{D} (ENTER(Z);return(TRIGGERED_TOKEN); )
(T){Y}{P}{E} (ENTER(Z);return(TYPE_TOKEN); )
(V){E}{R}{T}{E}{X} (ENTER(Z);return(VERTEX_TOKEN); )
"and"|"AND" (ENTER(Z);return(AND_TOKEN); )
"or"|"OR" (ENTER(Z);return(OR_TOKEN); )
"xor"|"XOR" (ENTER(Z);return(XOR_TOKEN); )
">=" (ENTER(Z);return(GREATER_THAN_OR_EQUAL); )
"<=" (ENTER(Z);return(LESS_THAN_OR_EQUAL); )

```

```

"/="|"~="      {ENTER(Z);return(INEQUALITY); }
"->"          {ENTER(Z);return(ARROW); }
"="           {ENTER(Z);return('='); }
"+"          {ENTER(Z);return('+'); }
"-"          {ENTER(Z);return('-'); }
"*"          {ENTER(Z);return('*'); }
"/"          {ENTER(Z);return('/'); }
"&"          {ENTER(Z);return('&'); }
"("          {ENTER(Z);return('('); }
")"          {ENTER(Z);return(')'); }
"["          {ENTER(Z);return('['); }
"]"          {ENTER(Z);return(']'); }
":"          {ENTER(Z);return(':'); }
","          {ENTER(Z);return(','); }
"."          {ENTER(Z);return('.'); }
"|"          {ENTER(Z);return('|'); }
">"          {ENTER(Z);return('>'); }
"<"          {ENTER(Z);return('<'); }
"mod"|"MOD"   {ENTER(Z);return(MOD_TOKEN); }
"rem"|"REM"   {ENTER(Z);return(REM_TOKEN); }
"***"|"EXP"|"exp" {ENTER(Z);return(EXP_TOKEN); }
"abs"|"ABS"   {ENTER(Z);return(ABS_TOKEN); }
"not"|"NOT"   {ENTER(Z);return(NOT_TOKEN); }
{T}{R}{U}{E} {ENTER(Z);return(TRUE); }
{F}{A}{L}{S}{E} {ENTER(Z);return(FALSE); }
<IDENT>\'     {ENTER(Z);return(''); }
{Letter}{Alpha}* {ENTER(IDENT);

    strlen(query_lex_dfa.yytext,length);

    the_id_token(1..length) := query_lex_dfa.yytext;

    return(IDENTIFIER); }
{Quote}{StrLit}*{Quote} {ENTER(Z);return(STRING_LITERAL); }
{Int}                   {ENTER(Z);return(INTEGER_LITERAL); }
{Int}" Cant {Int}       {ENTER(Z);return(REAL_LITERAL); }
{" Cant {Text} Cant "   {ENTER(Z);return(TEXT_TOKEN); }
[\n]                    {NULL;}
[ \t]                   {NULL;}--ignore spaces and tabs

%%
with query_Tokens,query_aux_pkg;
use query_Tokens,Text_io,query_aux_pkg;

package query_Lex is
    function yylex return token;
end query_Lex;

package body query_Lex is
    ##
end query_Lex;

```

D. QUERY.Y

```
-- *** File      : query.y
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

%token '(' ')' '[' ']' ':' ',' ' ' '.' '|'
%token ARROW
%token TRUE FALSE
%token ADA_TOKEN AXIOMS_TOKEN
%token BY_ALL_TOKEN REQ_BY_TOKEN BY_SOME_TOKEN
%token CALL_PERIOD_TOKEN CONTROL_TOKEN
%token CONSTRAINTS_TOKEN
%token DESCRIPTION_TOKEN DATA_TOKEN
%token END_TOKEN EDGE_TOKEN EXCEPTION_TOKEN
%token EXCEPTIONS_TOKEN EXECUTION_TOKEN
%token FINISH_TOKEN
%token GENERIC_TOKEN GRAPH_TOKEN
%token HOURS_TOKEN
%token IF_TOKEN IMPLEMENTATION_TOKEN
%token INPUT_TOKEN INITIALLY_TOKEN
%token KEYWORDS_TOKEN
%token MAXIMUM_TOKEN MINIMUM_TOKEN
%token MICROSEC_TOKEN MIN_TOKEN MS_TOKEN
%token MOD_TOKEN NOT_TOKEN
%token OPERATOR_TOKEN OR_TOKEN
%token OUTPUT_TOKEN
%token PERIOD_TOKEN
%token RESET_TOKEN RESPONSE_TOKEN
%token SPECIFICATION_TOKEN SEC_TOKEN
%token START_TOKEN STATES_TOKEN STOP_TOKEN
%token STREAM_TOKEN
%token TIME_TOKEN TIMER_TOKEN
%token TYPE_TOKEN TRIGGERED_TOKEN
%token VERTEX_TOKEN
%token WITHIN_TOKEN
%token TEXT_TOKEN
%token IDENTIFIER
%token STRING_LITERAL
%token INTEGER_LITERAL
%token REAL_LITERAL

--operator precedences
%left AND_TOKEN OR_TOKEN XOR_TOKEN LOGICAL_OPERATOR
%left '<' '>' '=' GREATER_THAN_OR_EQUAL LESS_THAN_OR_EQUAL INEQUALITY
RELATIONAL_OPERATOR
%left '+' '-' '&'amp;' BINARY_ADDING_OPERATOR
%left UNARY_ADDING_OPERATOR
%left '*' '/' MOD_TOKEN REM_TOKEN MULTIPLYING_OPERATOR
%left EXP_TOKEN ABS_TOKEN NOT_TOKEN HIGHEST_PRECEDENCE_OPERATOR
```

```

%start start_symbol

(
  subtype yystype is integer;
)

%%

start_symbol:
    psdl
    ;
psdl:
    psdl
    component
    |
    ;
component:
    data_type
    |
    operator
    ;
data_type:
    TYPE_TOKEN
    {
      query_is_adt := True;
    }
    IDENTIFIER
    {
      null;
    }
    type_spec type_impl
    ;
type_spec:
    SPECIFICATION_TOKEN optional_generic_param
optional_type_decl
    op_spec_list functionality END_TOKEN
    ;
optional_generic_param:
    GENERIC_TOKEN
    {
      query_is_generic := True;
      gen_par := True;
    }
    list_of_type_decl
    |
    ;
optional_type_decl:
    list_of_type_decl
    |
    ;
op_spec_list:

```



```

op_spec_list
OPERATOR_TOKEN
IDENTIFIER
{
tmp_op_name(1..length):=the_id_token(1..length);
len_name:=length;
}
operator_spec
{
    if query_is_adt then
        query_aux_pkg.Put_adt_op(tmp_op_name, len_name,
            has_inputs, has_outputs, query_adt_op_ptr);
        has_inputs:= False;
        has_outputs:= False;
        in_par:=False;
        out_par:=False;
    end if;
}
|
;
operator:
OPERATOR_TOKEN
{
    query_is_operator := True;
}
IDENTIFIER
{
    null;
}
operator_spec operator_impl
;
operator_spec:
SPECIFICATION_TOKEN
interface
functionality
END_TOKEN
;
interface:
interface attribute reqmts_trace
|
;
attribute:
GENERIC_TOKEN
{
query_is_generic := True;
gen_par := True;
}
list_of_type_decl
|
INPUT_TOKEN
{

```

```

query_has_in_param := True;
has_inputs      := True;
in_par := True;
gen_par := False;
}
list_of_type_decl
|
OUTPUT_TOKEN
{
    query_has_out_param := True;
    has_outputs      := True;
    in_par := False;
    out_par := True;
    gen_par := False;
}
list_of_type_decl
|
STATES_TOKEN
list_of_type_decl
INITIALLY_TOKEN
initial_expression_list
|
EXCEPTIONS_TOKEN
id_list
|
MAXIMUM_TOKEN EXECUTION_TOKEN TIME_TOKEN time
;
list_of_type_decl:
list_of_type_decl ',' type_decl
|
type_decl
;
type_decl:
id_list
(
    if gen_par then
        query_aux_pkg.Put_gen_parameter(the_id_token,
                                         " ", length, 1, query_gen_par_ptr);
    end if;
    if in_par then
        if out_par then
            null;
        else
            query_aux_pkg.Put_in_parameter(the_id_token,
                                           length, query_in_par_ptr);
        end if;
    end if;

    if out_par then
        query_aux_pkg.Put_out_parameter(the_id_token,
                                         length, query_out_par_ptr);
    end if;
)

```

```

        end if;
    }
    ':'
type_name
(
    if in_par then
        if out_par then
            null;
        else
            query_aux_pkg.Put_in_parameter_val(the_id_token,
                                                length,query_in_par_val_ptr);
        end if;
    end if;

    if out_par then
        query_aux_pkg.Put_out_parameter_val(the_id_token,
                                            length,query_out_par_val_ptr);
    end if;
}
;

type_name:
IDENTIFIER
 '['
list_of_type_decl
 ']'
|
IDENTIFIER
;

id_list:
id_list
 ','
IDENTIFIER
|
IDENTIFIER
;

reqmts_trace:
REQ_BY_TOKEN id_list
|
;

functionality:
keywords informal_desc formal_desc
;

keywords:
KEYWORDS_TOKEN
(
    out_par := False;
    gen_par := False;--in case no in/out parameters
)
id_list
|
;

```

```

informal_desc:
    DESCRIPTION_TOKEN TEXT_TOKEN
    |
    ;
formal_desc:
    AXIOMS_TOKEN TEXT_TOKEN
    |
    ;
type_impl:
    IMPLEMENTATION_TOKEN ADA_TOKEN IDENTIFIER
    {
        queryname.name(1..length):=the_id_token(1..length);
        queryname.len:=length;
    }
    END_TOKEN
    |
    IMPLEMENTATION_TOKEN type_name op_impl_list END_TOKEN
    ;
op_impl_list:
    op_impl_list OPERATOR_TOKEN IDENTIFIER operator_impl
    |
    ;
operator_impl:
    IMPLEMENTATION_TOKEN ADA_TOKEN IDENTIFIER
    {
        queryname.name(1..length):=the_id_token(1..length);
        queryname.len:=length;
    }
    END_TOKEN
    |
    IMPLEMENTATION_TOKEN psdl_impl END_TOKEN
    ;
psdl_impl:
    data_flow_diagram streams timers control_constraints
informal_desc
    ;
data_flow_diagram:
    GRAPH_TOKEN vertex_list edge_list
    ;
vertex_list:
    vertex_list VERTEX_TOKEN op_id optional_time
    |
    ;
edge_list:
    edge_list EDGE_TOKEN IDENTIFIER
    optional_time op_id ARROW op_id
    |
    ;
op_id:
    IDENTIFIER opt_arg
    ;

```

```

opt_arg:
    '(' optional_id_list
    '|' optional_id_list ')'
    |
    ;
optional_id_list:
    id_list
    |
    ;
optional_time:
    ':' time
    |
    ;
streams:
    DATA_TOKEN STREAM_TOKEN list_of_type_decl
    |
    ;
timers:
    TIMER_TOKEN id_list
    |
    ;
control_constraints:
    CONTROL_TOKEN CONSTRAINTS_TOKEN constraints
    ;
constraints:
    constraints OPERATOR_TOKEN IDENTIFIER
    opt_trigger opt_period opt_finish_within
    opt_mcp opt_mrt constraint_options
    |
    OPERATOR_TOKEN IDENTIFIER
    opt_trigger opt_period opt_finish_within
    opt_mcp opt_mrt
    ;
constraint_options:
    constraint_options OUTPUT_TOKEN id_list IF_TOKEN
    expression reqmts_trace
    |
    constraint_options EXCEPTION_TOKEN IDENTIFIER
    opt_if_predicate reqmts_trace
    |
    constraint_options timer_op IDENTIFIER
    opt_if_predicate reqmts_trace
    |
    ;
opt_trigger:
    TRIGGERED_TOKEN trigger opt_if_predicate reqmts_trace
    |
    ;
trigger:
    BY_ALL_TOKEN id_list
    |

```

```

        BY_SOME_TOKEN id_list
        |
        ;
opt_period:
        PERIOD_TOKEN time reqmts_trace
        |
        ;
opt_finish_within:
        FINISH_TOKEN WITHIN_TOKEN time reqmts_trace
        |
        ;
opt_mcp:
        MINIMUM_TOKEN CALL_PERIOD_TOKEN time reqmts_trace
        |
        ;
opt_mrt:
        max_resp_time time reqmts_trace
        |
        ;
max_resp_time:
        MAXIMUM_TOKEN RESPONSE_TOKEN TIME_TOKEN
        ;
timer_op:
        RESET_TOKEN
        |
        START_TOKEN
        |
        STOP_TOKEN
        ;
opt_if_predicate:
        IF_TOKEN expression
        |
        ;
initial_expression_list:
        initial_expression_list ',' initial_expression
        |
        initial_expression
        ;
initial_expression:
        TRUE
        |
        FALSE
        |
        INTEGER_LITERAL
        |
        REAL_LITERAL
        |
        STRING_LITERAL
        |
        IDENTIFIER
        |

```

```

        type_name '.' IDENTIFIER
        |
        type_name '.' IDENTIFIER '(' initial_expression_list ')'
        |
        '(' initial_expression_list ')'
        |
        initial_expression log_op initial_expression %prec
logical_operator
        |
        initial_expression rel_op initial_expression %prec
relational_operator
        |
        '-' initial_expression %prec unary_adding_operator
        |
        '+' initial_expression %prec unary_adding_operator
        |
        initial_expression bin_add_op initial_expression %prec
multiplying_operator
        |
        initial_expression bin_mul_op initial_expression %prec
multiplying_operator
        |
        initial_expression EXP_TOKEN initial_expression %prec
highest_precedence_operator
        |
        NOT_TOKEN initial_expression %prec
highest_precedence_operator
        |
        ABS_TOKEN initial_expression %prec
highest_precedence_operator
        ;
log_op:
        AND_TOKEN
        |
        OR_TOKEN
        |
        XOR_TOKEN
        ;
rel_op:
        '<' | '>' | '=' | GREATER_THAN_OR_EQUAL
        |
        LESS_THAN_OR_EQUAL | INEQUALITY
        ;
bin_add_op:
        '+' | '-' | '&'
        ;
bin_mul_op:
        '*' | '/' | MOD_TOKEN | REM_TOKEN
        ;
time:
        time_number MICROSEC_TOKEN

```

```

        |
        time_number MS_TOKEN
        |
        time_number SEC_TOKEN
        |
        time_number MIN_TOKEN
        |
        time_number HOURS_TOKEN
        ;
time_number:
    INTEGER_LITERAL
    ;
expression_list:
    expression_list ',' expression
    |
    expression
    ;
expression:
    TRUE | FALSE | INTEGER_LITERAL | time | REAL_LITERAL
    | STRING_LITERAL | IDENTIFIER
    | type_name '.' IDENTIFIER
    | type_name '.' IDENTIFIER '(' expression_list ')'
    | '(' expression_list ')'
    | expression log_op expression %prec logical_operator
    | expression rel_op expression %prec relational_operator
    | '-' expression %prec unary_adding_operator
    | '+' expression %prec unary_adding_operator
    | expression bin_add_op expression %prec
unary_adding_operator
    | expression bin_mul_op expression %prec
multiplying_operator
    | expression EXP_TOKEN expression %prec
highest_precedence_operator
    | NOT_TOKEN expression %prec highest_precedence_operator
    | ABS_TOKEN expression %prec highest_precedence_operator
    ;

%%

with Text_io, query_tokens, query_goto, query_shift_reduce, query_lex,
     query_aux_pkg;
use Text_io, query_tokens, query_goto, query_shift_reduce, query_lex,
     query_aux_pkg;

package queryparser is

    procedure Yyparse;

    echo          : boolean := false;
    number_of_errors : Natural := 0;

```



```

in_par      : Boolean:=False;
out_par     : Boolean:=False;
gen_par     : Boolean:=False;
has_inputs  : Boolean:=False;
has_outputs : Boolean:=False;
tmp_op_name : String(1..80);
len_name    : Integer:=0;

end queryparser;

package body queryparser is

  procedure Yerror(s : in string := "syntax error") is
  begin
    number_of_errors := number_of_errors + 1;
    put ("*** ");
    put_line(s);
  end Yerror;

  ##%procedure_parse

end queryparser;

```

E. COMPONENT_AUX_PKG

1. COMPONENT_AUX_S.A

```
-- *** File      : component_aux_s.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

with Text_Io;
use Text_Io;

package component_aux_pkg is

    the_id_token      : String(1..80);
    length             : Integer:=1;
    has_in_param       : Boolean := False;
    has_out_param      : Boolean := False;
    is_operator        : Boolean := False;
    is_adt             : Boolean := False;
    is_generic         : Boolean := False;

    -- Global list constructors
    -- These variables and structures will hold the necessary info
    -- to construct the output file(s)

    type with_name is
        record
            name : String(1..80);
            len  : Integer:=0;
        end record;

    type in_parameter_list;
    type in_parameter_ptr is access in_parameter_list;
    type in_parameter_list is
        record
            name : String(1..80);
            len  : Integer:=0;
            link : in_parameter_ptr;
        end record;

    type out_parameter_list;
    type out_parameter_ptr is access out_parameter_list;
    type out_parameter_list is
        record
            name : String(1..80);
            len  : Integer:=0;
            link : out_parameter_ptr;
        end record;
```

```

type in_parameter_val_list;
type in_parameter_val_ptr is access in_parameter_val_list;

type in_parameter_val_list is
    record
        name : String(1..80);
        len  : Integer:=0;
        link : in_parameter_val_ptr;
    end record;

type out_parameter_val_list;
type out_parameter_val_ptr is access out_parameter_val_list;
type out_parameter_val_list is
    record
        name : String(1..80);
        len  : Integer:=0;
        link : out_parameter_val_ptr;
    end record;

type gen_parameter_list;
type gen_parameter_ptr is access gen_parameter_list;
type gen_parameter_list is
    record
        param: String(1..80);
        val  : String(1..80);
        link : gen_parameter_ptr;
    end record;

type adt_operator_list;
type adt_operator_ptr is access adt_operator_list;
type adt_operator_list is
    record
        op_name      : String(1..80);
        has_inputs   : Boolean:=False;
        has_outputs  : Boolean:=False;
        op_in_par    : in_parameter_ptr;
        op_in_par_val : in_parameter_val_ptr;
        op_out_par    : out_parameter_ptr;
        op_out_par_val: out_parameter_val_ptr;
        link         : adt_operator_ptr;
    end record;

withname      : with_name;
in_par_ptr    : in_parameter_ptr;
out_par_ptr   : out_parameter_ptr;
in_par_val_ptr : in_parameter_val_ptr;
out_par_val_ptr : out_parameter_val_ptr;
gen_par_ptr   : gen_parameter_ptr;
adt_op_ptr    : adt_operator_ptr;

```

```

procedure Put_in_parameter(Data: in String;
                           Len : in Integer;
                           List: in out in_parameter_ptr);

procedure Put_out_parameter(Data: in String;
                            Len : in Integer;
                            List: in out out_parameter_ptr);

procedure Put_in_parameter_val(Data: in String;
                               Len : in Integer;
                               List: in out in_parameter_val_ptr);

procedure Put_out_parameter_val(Data: in String;
                                Len : in Integer;
                                List: in out out_parameter_val_ptr);

procedure Put_gen_parameter(Parm : in String;
                            Val  : in String;
                            P_Len: in Integer;
                            V_Len: in Integer;
                            List : in out gen_parameter_ptr);

procedure Put_adt_op(name: in String;
                    len : in Integer;
                    inputs,outputs : in Boolean;
                    List : in out adt_operator_ptr);

procedure strlen(s: in String; n: in out Integer);
end component_aux_pkg;

```

2. COMPONENT_AUX_B.A

```

-- *** File      : component_aux_b.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

```

```

with Text_IO;
use Text_IO;

```

```

package body component_aux_pkg is

```

```

    procedure Put_in_parameter (Data: in String;
                                Len : in Integer;
                                List: in out in_parameter_ptr) is
    begin
        if List = null then
            List := new in_parameter_list;
            List.name(1..Len):=Data(1..Len);

```

```

        List.len:=Len;
        List.link:=null;
    else
        Put_in_parameter(Data,Len,List.link);
    end if;
end Put_in_parameter;

```

```

procedure Put_out_parameter(Data: in String;
                           Len : in Integer;
                           List: in out out_parameter_ptr) is

begin
    if List = null then
        List := new out_parameter_list;
        List.name(1..Len):=Data(1..Len);
        List.len:=Len;
        List.link:=null;
    else
        Put_out_parameter(Data,Len,List.link);
    end if;
end Put_out_parameter;

```

```

procedure Put_in_parameter_val(Data: in String;
                              Len : in Integer;
                              List: in out in_parameter_val_ptr) is

begin
    if List = null then
        List := new in_parameter_val_list;
        List.name(1..Len):=Data(1..Len);
        List.len:=Len;
        List.link:=null;
    else
        Put_in_parameter_val(Data,Len,List.link);
    end if;
end Put_in_parameter_val;

```

```

procedure Put_out_parameter_val(Data: in String;
                                Len : in Integer;
                                List: in out out_parameter_val_ptr) is

begin
    if List = null then
        List := new out_parameter_val_list;
        List.name(1..Len):=Data(1..Len);
        List.len:=Len;
        List.link:=null;
    else

```

```

        Put_out_parameter_val(Data,Len,List.link);
    end if;
end Put_out_parameter_val;

```

```

procedure Put_gen_parameter(Parm : in String;
                           Val  : in String;
                           P_Len: in Integer;
                           V_Len: in Integer;
                           List : in out gen_parameter_ptr) is
begin
    if List = null then
        List := new gen_parameter_list;
        List.param(1..P_Len):=Parm(1..P_Len);
        List.val(1..V_Len):=Val(1..V_Len);
        List.link:=null;
    else
        Put_gen_parameter(Parm,Val,P_Len,V_Len,List.link);
    end if;
end Put_gen_parameter;

```

```

procedure Put_adt_op  ( name: in String;
                       len  : in Integer;
                       inputs,outputs : in Boolean;
                       List: in out adt_operator_ptr) is
begin
    if List = null then
        List := new adt_operator_list;
        List.op_name(1..len):= name(1..len);
        List.has_inputs:=inputs;
        List.has_outputs:=outputs;
        List.op_in_par:= in_par_ptr;
        List.op_in_par_val:= in_par_val_ptr;
        List.op_out_par:=out_par_ptr;
        List.op_out_par_val:=out_par_val_ptr;
        List.link:=null;
    else
        Put_adt_op(name,len,inputs,outputs,List.link);
    end if;
    in_par_ptr:=null;
    in_par_val_ptr:=null;
    out_par_ptr:=null;
    out_par_val_ptr:=null;
end Put_adt_op;

```

```

-----

procedure strlen(s: in String; n: in out Integer) is
    I : Integer := 1;
    char:Character:='x';
    endstr:Boolean:=False;

begin
    while endstr=False loop
        begin
            char:=s(I);
            I:=I+1;
            exception
                when constraint_error => endstr:=True;
        end;
    end loop;
    n:= I-1;
end strlen;

-----

end component_aux_pkg;

```

F. COMPONENT_COMPOSE_PKG

1. COMPONENT_COMPOSE_S.A

```
-- *** File      : component_compose_s.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****
```

```
with text_io,component_aux_pkg;
use text_io,component_aux_pkg;
```

```
package component_compose_pkg is
```

```
    inpar_ptr    : in_parameter_ptr;
    outpar_ptr    : out_parameter_ptr;
    inparv_ptr    : in_parameter_val_ptr;
    outparv_ptr   : out_parameter_val_ptr;
    genpar_ptr    : gen_parameter_ptr;
    adtop_ptr     : adt_operator_ptr;
    opin_par      : in_parameter_ptr;
    opin_parv     : in_parameter_val_ptr;
    opout_par     : out_parameter_ptr;
    opout_parv    : out_parameter_val_ptr;
    outfile       : file_type;
    genfile       : file_type;
    actual_val     : string (1..80);
    gen_found     : Boolean:=False;
```

```
    procedure non_generic_operator;
    procedure generic_operator;
    procedure non_generic_adt;
    procedure generic_adt;
```

```
end component_compose_pkg;
```

2. COMPONENT_COMPOSE_B.A

```
-- *** File      : component_compose_b.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****
```

```
with text_io,component_compose_pkg;
use text_io,component_compose_pkg;
```

```
package body component_compose_pkg is
```

```
    procedure non_generic_operator is
```



```

begin
  create(outfile,mode=>out_file,name=>"outfile");
  create(genfile,mode=>out_file,name=>"genfile");
  put(genfile,"      ");
  put(outfile,"with ");
  put(outfile,withname.name(1..withname.len));
  put(outfile,";");
  new_line(outfile,2);
  put(outfile,"package ");
  put(outfile,withname.name(1..(withname.len-3)));
  put(outfile,"_pkg is");
  new_line(outfile,2);
  put(outfile,"  procedure ");
  put(outfile,withname.name(1..(withname.len-3)));

  if has_in_param then
    put(outfile," (");
    inpar_ptr:= in_par_ptr;
    inparv_ptr:= in_par_val_ptr;
    while inpar_ptr /= null loop
      Put(outfile,inpar_ptr.name);
      inpar_ptr:=inpar_ptr.link;
      Put(outfile," : in ");
      Put(outfile,inparv_ptr.name(1..inparv_ptr.len) );
      inparv_ptr:= inparv_ptr.link;
      if inpar_ptr /= null then
        Put(outfile,";");
        new_line(outfile);set_col(outfile,13);
      else
        if has_out_param then
          Put(outfile,";");
          new_line(outfile);
          set_col(outfile,13);
          exit;
        else
          Put(outfile," )");
          new_line(outfile);
        end if;
      end if;
    end loop;
  end if;

  if has_out_param then
    if has_in_param then
      null;
    else
      put(outfile,"(");
    end if;
    outpar_ptr:= out_par_ptr;
    outparv_ptr:= out_par_val_ptr;
    while outpar_ptr /= null loop

```

```

    Put(outfile,outpar_ptr.name(1..outpar_ptr.len) );
    outpar_ptr:= outpar_ptr.link;
    Put(outfile," : out ");
    Put(outfile,outparv_ptr.name(1..outparv_ptr.len) );
    outparv_ptr:=outparv_ptr.link;
    if outpar_ptr /= null then
        Put(outfile,";");
        new_line(outfile);set_col(outfile,13);
    else
        Put(outfile," )");
        new_line(outfile);
    end if;
end loop;
end if;

new_line(outfile);
put(outfile," renames ");
put(outfile,withname.name);
put(outfile,".");
put(outfile,withname.name(1..(withname.len-3)));
put(outfile,";");
new_line(outfile,2);
put(outfile,"end ");
put(outfile,withname.name(1..(withname.len-3)));
put(outfile,"_pkg;");
close(outfile);
end non_generic_operator;

```

```

procedure generic_operator is

```

```

begin
    create(outfile,mode=>out_file,name=>"outfile");
    create(genfile,mode=>out_file,name=>"genfile");
    put(genfile,"generic");
    put(outfile,"with ");
    put(outfile,withname.name(1..withname.len));
    put(outfile,";");
    new_line(outfile,2);
    put(outfile,"package ");
    put(outfile,withname.name(1..(withname.len-3)));
    put(outfile,"_pkg is");
    new_line(outfile,2);

```

--At this point we need actual values of generic units for instantiation.

--This part of code is commented out for reference.

```

--genpar_ptr:=gen_par_ptr;
--while genpar_ptr /= null loop
    --put ("Enter actual parameter for generic <");
    --put (genpar_ptr.name(1..genpar_ptr.len) );

```

```

--put ("> : ");
--get_line(actual_val,length);
--component_aux_pkg.Put_gen_parameter_val
      (actual_val,length,gen_par_val_ptr);
--new_line;
--genpar_ptr:=genpar_ptr.link;
--end loop;

--now continue with instantiation

put(outfile," package tmp_");
put(outfile,withname.name(1..(withname.len-3)));
put(outfile,"_pkg is new ");
put(outfile,withname.name(1..withname.len));
put(outfile," ( ");
genpar_ptr:=gen_par_ptr;
while genpar_ptr /= null loop
  Put(outfile,genpar_ptr.val);
  if genpar_ptr.link /= null then
    Put(outfile,", ");
  end if;
  genpar_ptr:=genpar_ptr.link;
end loop;
put(outfile," );");
new_line(outfile,2);

--now rename

put(outfile," procedure ");
put(outfile,withname.name(1..(withname.len-3)));

if has_in_param then
  put(outfile," (");
  inpar_ptr:= in_par_ptr;
  inparv_ptr:= in_par_val_ptr;
  while inpar_ptr /= null loop
    Put(outfile,inpar_ptr.name(1..inpar_ptr.len) );
    Put(outfile," : in ");
    inpar_ptr:=inpar_ptr.link;

    genpar_ptr:=gen_par_ptr;
    while genpar_ptr /= null loop
      if inparv_ptr.name = genpar_ptr.param then
        Put(outfile,genpar_ptr.val);
        gen_found:=True;
        exit;
      end if;
      genpar_ptr:=genpar_ptr.link;
    end loop;
  end loop;
  if gen_found=False then
    Put(outfile,inparv_ptr.name(1..inparv_ptr.len) );

```

```

end if;
gen_found:=False;

inparv_ptr:= inparv_ptr.link;
if inpar_ptr /= null then
    Put(outfile,";");
    new_line(outfile);
    set_col(outfile,13);
else
    if has_out_param then
        Put(outfile,";");
        new_line(outfile);
        set_col(outfile,13);
        exit;
    else
        Put(outfile," ");
        new_line(outfile);
    end if;
end if;
end loop;
end if;

if has_out_param then
    if has_in_param then
        null;
    else
        put(outfile,"(");
    end if;
    outpar_ptr:= out_par_ptr;
    outparv_ptr:= out_par_val_ptr;
    while outpar_ptr /= null loop
        Put(outfile,outpar_ptr.name(1..outpar_ptr.len) );
        outpar_ptr:= outpar_ptr.link;
        Put(outfile," : out ");
    end loop;

    genpar_ptr:=gen_par_ptr;
    while genpar_ptr /= null loop
        if outparv_ptr.name = genpar_ptr.param then
            Put(outfile,genpar_ptr.val);
            gen_found:=True;
            exit;
        end if;
        genpar_ptr:=genpar_ptr.link;
    end loop;
    if gen_found=False then
        Put(outfile,outparv_ptr.name(1..outparv_ptr.len) );
    end if;
    gen_found:=False;

    outparv_ptr:=outparv_ptr.link;
    if outpar_ptr /= null then

```

```

        Put(outfile, ";");
        new_line(outfile);
        set_col(outfile, 13);
    else
        Put(outfile, " ");
        new_line(outfile);
    end if;
end loop;
end if;

new_line(outfile);
put(outfile, " renames tmp_");
put(outfile, withname.name(1..(withname.len-3)));
put(outfile, "_pkg.");
put(outfile, withname.name(1..(withname.len-3)));
put(outfile, ";");
new_line(outfile, 2);
put(outfile, "end ");
put(outfile, withname.name(1..(withname.len-3)));
put(outfile, "_pkg;");
close(outfile);
end generic_operator;

```

```

procedure non_generic_adt is
begin
    create(outfile, mode=>out_file, name=>"outfile");
    create(genfile, mode=>out_file, name=>"genfile");
    put(genfile, " ");
    put(outfile, "with ");
    put(outfile, withname.name(1..withname.len));
    put(outfile, ";");
    new_line(outfile, 2);
    put(outfile, "package ");
    put(outfile, withname.name(1..(withname.len-3)));
    put(outfile, "_pkg is");
    new_line(outfile, 2);

    adtop_ptr:=adt_op_ptr;
    while adtop_ptr /= null loop
        put(outfile, " procedure ");
        Put(outfile, adtop_ptr.op_name);

        if adtop_ptr.has_inputs then
            put(outfile, " (");
            opin_par:=adtop_ptr.op_in_par;
            opin_parv:=adtop_ptr.op_in_par_val;
            while opin_par /= null loop
                put(outfile, opin_par.name);
            end loop;
        end if;
    end loop;
end non_generic_adt;

```

```

    opin_par:=opin_par.link;
    Put(outfile," : in ");
    --generic par check is to go here
    put(outfile,opin_parv.name);
    opin_parv:=opin_parv.link;
    if opin_par /= null then
        Put(outfile,";");
        new_line(outfile);
        set_col(outfile,13);
    else
        if adtop_ptr.has_outputs then
            Put(outfile,";");
            new_line(outfile);
            set_col(outfile,13);
        exit;
        else
            Put(outfile," )");
            new_line(outfile);
        end if;
    end if;
end loop;
end if;

if adtop_ptr.has_outputs then
    if adtop_ptr.has_inputs then
        null;
    else
        put(outfile,"(");
    end if;
    opout_par:=adtop_ptr.op_out_par;
    opout_parv:=adtop_ptr.op_out_par_val;
    while opout_par /= null loop
        Put(outfile,opout_par.name);
        opout_par:=opout_par.link;
        Put(outfile," : out ");
        put(outfile,opout_parv.name);
        opout_parv:=opout_parv.link;
        if opout_par /= null then
            Put(outfile,";");
            new_line(outfile);
            set_col(outfile,13);
        else
            Put(outfile," )");
            new_line(outfile);
        end if;
    end loop;
end if;
new_line(outfile);
put(outfile," renames ");
put(outfile,withname.name);
put(outfile,".");

```

```

    put(outfile,adtop_ptr.op_name);
    put(outfile,";");
    new_line(outfile,2);
    adtop_ptr:=adtop_ptr.link;
end loop;

    put(outfile,"end ");
    put(outfile,withname.name(1..(withname.len-3)));
    put(outfile,"_pkg;");
    close(outfile);

end non_generic_adt;

```

```

procedure generic_adt is

```

```

begin
    create(outfile,mode=>out_file,name=>"outfile");
    create(genfile,mode=>out_file,name=>"genfile");
    put(genfile,"generic");
    put(outfile,"with ");
    put(outfile,withname.name(1..withname.len));
    put(outfile,";");
    new_line(outfile,2);
    put(outfile,"package ");
    put(outfile,withname.name(1..(withname.len-3)));
    put(outfile,"_pkg is");
    new_line(outfile,2);

    put(outfile," package tmp_");
    put(outfile,withname.name(1..(withname.len-3)));
    put(outfile,"_pkg is new ");
    put(outfile,withname.name(1..withname.len));
    put(outfile," ( ");
    genpar_ptr:=gen_par_ptr;
    while genpar_ptr /= null loop
        Put(outfile,genpar_ptr.val);
        if genpar_ptr.link /= null then
            Put(outfile,", ");
        end if;
        genpar_ptr:=genpar_ptr.link;
    end loop;
    put(outfile," );");
    new_line(outfile,2);

    --now rename

    adtop_ptr:=adt_op_ptr;
    while adtop_ptr /= null loop
        put(outfile," procedure ");

```

```

Put(outfile,adtop_ptr.op_name);

if adtop_ptr.has_inputs then
    put(outfile," (");
    opin_par:=adtop_ptr.op_in_par;
    opin_parv:=adtop_ptr.op_in_par_val;
    while opin_par /= null loop
        put(outfile,opin_par.name);
        opin_par:=opin_par.link;
        Put(outfile," : in ");
        --generic par check is to go here for generic proc
        genpar_ptr:=gen_par_ptr;
        while genpar_ptr /= null loop
            if opin_parv.name=genpar_ptr.param then
                Put(outfile,genpar_ptr.val);
            gen_found:=True;
            exit;
        end if;
        genpar_ptr:=genpar_ptr.link;
    end loop;
    if gen_found=False then
        put(outfile,opin_parv.name);
    end if;
    gen_found:=False;

    opin_parv:=opin_parv.link;
    if opin_par /= null then
        Put(outfile,"");
        new_line(outfile);
        set_col(outfile,13);
    else
        if adtop_ptr.has_outputs then
            Put(outfile,"");
            new_line(outfile);
            set_col(outfile,13);
            exit;
        else
            Put(outfile," )");
            new_line(outfile);
        end if;
    end if;
end loop;
end if;

if adtop_ptr.has_outputs then
    if adtop_ptr.has_inputs then
        null;
    else
        put(outfile," (");
    end if;
    opout_par:=adtop_ptr.op_out_par;

```



```

    opout_parv:=adtop_ptr.op_out_par_val;
    while opout_par /= null loop
        Put(outfile,opout_par.name);
        opout_par:=opout_par.link;
        Put(outfile," : out ");
        --generic par check
        genpar_ptr:=gen_par_ptr;
        while genpar_ptr /= null loop
            if opout_parv.name=genpar_ptr.param then
                Put(outfile,genpar_ptr.val);
                gen_found:=True;
                exit;
            end if;
            genpar_ptr:=genpar_ptr.link;
        end loop;
        if gen_found=False then
            put(outfile,opout_parv.name);
        end if;
        gen_found:=False;

        opout_parv:=opout_parv.link;
        if opout_par /= null then
            Put(outfile,";");
            new_line(outfile);
            set_col(outfile,13);
        else
            Put(outfile," ");
            new_line(outfile);
        end if;
    end loop;
    end if;
    new_line(outfile);
    put(outfile," renames tmp_");
    put(outfile,withname.name(1..(withname.len-3)));
    put(outfile,"_pkg.");
    put(outfile,adtop_ptr.op_name);
    put(outfile,";");
    new_line(outfile,2);
    adtop_ptr:=adtop_ptr.link;
end loop;

put(outfile,"end ");
put(outfile,withname.name(1..(withname.len-3)));
put(outfile,"_pkg.");
close(outfile);

end generic_adt;

end component_compose_pkg;

```

G. COMPONENT_PARSE.A

```
-- *** File      : component_parse.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

with u_env, component_parser, psdl_lex_io, psdl_lex, text_io,
     component_aux_pkg, component_compose_pkg;
use u_env, component_parser, text_io, component_aux_pkg,
     component_compose_pkg;

procedure parse is

    last      : integer:=0;

begin
    if u_env.Argc = 2 then

        strlen(Argv(1).S, last);
        psdl_lex_io.open_input(Argv(1).S(1..last));
        psdl_lex_io.create_output;

        yyparse;

        psdl_lex_io.close_input;
        psdl_lex_io.close_output;

        if is_generic and is_operator then
            component_compose_pkg.generic_operator;
        elsif is_generic=False and is_operator then
            component_compose_pkg.non_generic_operator;
        elsif is_generic and is_adt then
            generic_adt;
        elsif is_generic=False and is_adt then
            non_generic_adt;
        end if;

    else
        put_line("Wrong number of arguments");
    end if;

end parse;
```

H. QUERY_AUX_PKG

1. QUERY_AUX_S.A

```
-- *** File      : query_aux_s.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

with Text_Io;
use Text_Io;

package query_aux_pkg is

    the_id_token      : String(1..80);
    length            : Integer:=1;
    query_has_in_param : Boolean := False;
    query_has_out_param : Boolean := False;
    query_is_operator  : Boolean := False;
    query_is_adt       : Boolean := False;
    query_is_generic   : Boolean := False;

    -- Global list constructors
    -- These variables and structures will hold the necessary info
    -- to construct the output file(s)

    type query_name is
        record
            name : String(1..80);
            len  : Integer:=0;
        end record;

    type query_in_parameter_list;
    type query_in_parameter_ptr is access
        query_in_parameter_list;
    type query_in_parameter_list is
        record
            name : String(1..80);
            len  : Integer:=0;
            link : query_in_parameter_ptr;
        end record;

    type query_out_parameter_list;
    type query_out_parameter_ptr is access
        query_out_parameter_list;
    type query_out_parameter_list is
        record
            name : String(1..80);
```

```

        len : Integer:=0;
        link : query_out_parameter_ptr;
    end record;

type query_in_parameter_val_list;
type query_in_parameter_val_ptr is access
    query_in_parameter_val_list;
type query_in_parameter_val_list is
    record
        name : String(1..80);
        len : Integer:=0;
        link : query_in_parameter_val_ptr;
    end record;

type query_out_parameter_val_list;
type query_out_parameter_val_ptr is access
    query_out_parameter_val_list;
type query_out_parameter_val_list is
    record
        name : String(1..80);
        len : Integer:=0;
        link : query_out_parameter_val_ptr;
    end record;

type query_gen_parameter_list;
type query_gen_parameter_ptr is access
    query_gen_parameter_list;
type query_gen_parameter_list is
    record
        param: String(1..80);
        val : String(1..80);
        link : query_gen_parameter_ptr;
    end record;

type query_adt_operator_list;
type query_adt_operator_ptr is access
    query_adt_operator_list;
type query_adt_operator_list is
    record
        op_name      : String(1..80);
        has_inputs   : Boolean:=False;
        has_outputs  : Boolean:=False;
        op_in_par    : query_in_parameter_ptr;
        op_in_par_val : query_in_parameter_val_ptr;
        op_out_par   : query_out_parameter_ptr;
        op_out_par_val : query_out_parameter_val_ptr;
        link         : query_adt_operator_ptr;
    end record;

```

```

queryname           : query_name;
query_in_par_ptr    : query_in_parameter_ptr;
query_out_par_ptr   : query_out_parameter_ptr;
query_in_par_val_ptr : query_in_parameter_val_ptr;
query_out_par_val_ptr : query_out_parameter_val_ptr;
query_gen_par_ptr   : query_gen_parameter_ptr;
query_adt_op_ptr    : query_adt_operator_ptr;

procedure Put_in_parameter(Data: in String;
                           Len : in Integer;
                           List: in out query_in_parameter_ptr);

procedure Put_out_parameter(Data: in String;
                            Len : in Integer;
                            List: in out query_out_parameter_ptr);

procedure Put_in_parameter_val(Data: in String;
                               Len : in Integer;
                               List: in out query_in_parameter_val_ptr);

procedure Put_out_parameter_val(Data: in String;
                                Len : in Integer;
                                List: in out query_out_parameter_val_ptr);

procedure Put_gen_parameter(Parm: in String;
                            Val : in String;
                            P_Len: in Integer;
                            V_Len: in Integer;
                            List: in out query_gen_parameter_ptr);

procedure Put_adt_op(name: in String;
                    len : in Integer;
                    inputs,outputs : in Boolean;
                    List : in out query_adt_operator_ptr);

procedure strlen(s: in String; n: in out Integer);

end query_aux_pkg;

```

2. QUERY_AUX_B.A

```
-- *** File      : query_aux_b.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

with Text_Io;
use Text_Io;

package body query_aux_pkg is

    procedure Put_in_parameter (Data: in String;
                                Len : in Integer;
                                List: in out query_in_parameter_ptr) is

    begin
        if List = null then
            List := new query_in_parameter_list;
            List.name(1..Len) := Data(1..Len);
            List.len := Len;
            List.link := null;
        else
            Put_in_parameter(Data, Len, List.link);
        end if;
    end Put_in_parameter;

-----

    procedure Put_out_parameter(Data: in String;
                                Len : in Integer;
                                List: in out query_out_parameter_ptr) is

    begin
        if List = null then
            List := new query_out_parameter_list;
            List.name(1..Len) := Data(1..Len);
            List.len := Len;
            List.link := null;
        else
            Put_out_parameter(Data, Len, List.link);
        end if;
    end Put_out_parameter;

-----

    procedure Put_in_parameter_val(Data: in String;
                                    Len : in Integer;
                                    List: in out query_in_parameter_val_ptr) is

    begin
        if List = null then
```

```

        List := new query_in_parameter_val_list;
        List.name(1..Len) := Data(1..Len);
        List.len := Len;
        List.link := null;
    else
        Put_in_parameter_val(Data, Len, List.link);
    end if;
end Put_in_parameter_val;

```

```

procedure Put_out_parameter_val(Data: in String;
                                Len : in Integer;
                                List: in out query_out_parameter_val_ptr) is
begin
    if List = null then
        List := new query_out_parameter_val_list;
        List.name(1..Len) := Data(1..Len);
        List.len := Len;
        List.link := null;
    else
        Put_out_parameter_val(Data, Len, List.link);
    end if;
end Put_out_parameter_val;

```

```

procedure Put_gen_parameter(Parm : in String;
                            Val  : in String;
                            P_Len: in Integer;
                            V_Len: in Integer;
                            List : in out query_gen_parameter_ptr) is
begin
    if List = null then
        List := new query_gen_parameter_list;
        List.param(1..P_Len) := Parm(1..P_Len);
        List.val(1..V_Len) := Val(1..V_Len);
        List.link := null;
    else
        Put_gen_parameter(Parm, Val, P_Len, V_Len, List.link);
    end if;
end Put_gen_parameter;

```

```

procedure Put_adt_op(name: in String;
                    len : in Integer;
                    inputs, outputs : in Boolean;
                    List : in out query_adt_operator_ptr) is

```

```

begin
  if List = null then
    List := new query_adt_operator_list;
    List.op_name(1..len) := name(1..len);
    List.has_inputs:=inputs;
    List.has_outputs:=outputs;
    List.op_in_par:= query_in_par_ptr;
    List.op_in_par_val:= query_in_par_val_ptr;
    List.op_out_par:=query_out_par_ptr;
    List.op_out_par_val:=query_out_par_val_ptr;
    List.link:=null;
  else
    Put_adt_op(name,len,inputs,outputs,List.link);
  end if;
  query_in_par_ptr:=null;
  query_in_par_val_ptr:=null;
  query_out_par_ptr:=null;
  query_out_par_val_ptr:=null;
end Put_adt_op;

-----

procedure strlen(s: in String; n: in out Integer) is

  I : Integer := 1;
  char:Character:='x';
  endstr:Boolean:=False;

begin
  while endstr=False loop
    begin
      char:=s(I);
      I:=I+1;
    exception
      when constraint_error => endstr:=True;
    end;
  end loop;
  n:= I-1;
end strlen;

-----

end query_aux_pkg;

```


I. QUERY_COMPOSE_PKG

1. QUERY_COMPOSE_S.A

```
-- *** File      : query_compose_s.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****
```

```
with text_io,query_aux_pkg,component_aux_pkg;
use text_io,query_aux_pkg,component_aux_pkg;
```

```
package query_compose_pkg is
```

```
    inpar_ptr   : query_in_parameter_ptr;
    outpar_ptr  : query_out_parameter_ptr;
    inparv_ptr  : query_in_parameter_val_ptr;
    outparv_ptr : query_out_parameter_val_ptr;
    genpar_ptr  : gen_parameter_ptr;
    adtop_ptr   : query_adt_operator_ptr;
    r_adtop_ptr : adt_operator_ptr;
    opin_par    : query_in_parameter_ptr;
    opin_parv   : query_in_parameter_val_ptr;
    opout_par   : query_out_parameter_ptr;
    opout_parv  : query_out_parameter_val_ptr;
    outfile     : file_type;
    genfile     : file_type;
    actual_val   : string (1..80);
    gen_found   : Boolean:=False;
```

```
    procedure non_generic_operator;
    procedure generic_operator;
    procedure non_generic_adt;
    procedure generic_adt;
```

```
end query_compose_pkg;
```

2. QUERY_COMPOSE_B.A

```
-- *** File      : query_compose_b.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****
```

```
with text_io,query_compose_pkg;
use text_io,query_compose_pkg;
```

```
package body query_compose_pkg is
```

```
    procedure non_generic_operator is
```

```

begin
  create(outfile,mode=>out_file,name=>"outfile");
  create(genfile,mode=>out_file,name=>"genfile");
  put(genfile,"      ");
  put(outfile,"with ");
  put(outfile,withname.name(1..withname.len));
  put(outfile,";");
  new_line(outfile,2);
  put(outfile,"package ");
  put(outfile,queryname.name(1..queryname.len));
  put(outfile,"_pkg is");
  new_line(outfile,2);
  put(outfile,"  procedure ");
  put(outfile,queryname.name(1..queryname.len));

  if query_has_in_param then
    put(outfile," (");
    inpar_ptr:= query_in_par_ptr;
    inparv_ptr:= query_in_par_val_ptr;
    while inpar_ptr /= null loop
      Put(outfile,inpar_ptr.name);
      inpar_ptr:=inpar_ptr.link;
      Put(outfile," : in ");
      Put(outfile,inparv_ptr.name(1..inparv_ptr.len) );
      inparv_ptr:= inparv_ptr.link;
      if inpar_ptr /= null then
        Put(outfile,";");
        new_line(outfile); set_col(outfile,13);
      else
        if query_has_out_param then
          Put(outfile,";");
          new_line(outfile); set_col(outfile,13);
          exit;
        else
          Put(outfile," )");
          new_line(outfile);
        end if;
      end if;
    end loop;
  end if;

  if query_has_out_param then
    if query_has_in_param then
      null;
    else
      put(outfile,"(");
    end if;
    outpar_ptr:= query_out_par_ptr;
    outparv_ptr:= query_out_par_val_ptr;
    while outpar_ptr /= null loop
      Put(outfile,outpar_ptr.name(1..outpar_ptr.len) );

```

```

        outpar_ptr:= outpar_ptr.link;
        Put(outfile," : out ");
        Put(outfile,outparv_ptr.name(1..outparv_ptr.len));
        outparv_ptr:=outparv_ptr.link;
        if outpar_ptr /= null then
            Put(outfile,";");
            new_line(outfile);set_col(outfile,13);
        else
            Put(outfile," ");
            new_line(outfile);
        end if;
    end loop;
end if;

    new_line(outfile);
    put(outfile," renames ");
    put(outfile,withname.name);
    put(outfile,".");
    put(outfile,withname.name(1..(withname.len-3)));
    put(outfile,";");
    new_line(outfile,2);
    put(outfile,"end ");
    put(outfile,queryname.name(1..queryname.len));
    put(outfile,"_pkg;");
    close(outfile);
end non_generic_operator;

```

procedure generic_operator is

```

begin
    create(outfile,mode=>out_file,name=>"outfile");
    create(genfile,mode=>out_file,name=>"genfile");
    put(genfile,"generic");
    put(outfile,"with ");
    put(outfile,withname.name(1..withname.len));
    put(outfile,";");
    new_line(outfile,2);
    put(outfile,"package ");
    put(outfile,queryname.name(1..queryname.len));
    put(outfile,"_pkg is");
    new_line(outfile,2);

```

--At this point we need actual values of generic units for instantiation.

--now continue with instantiation

```

    put(outfile," package tmp_");
    put(outfile,queryname.name(1..queryname.len));
    put(outfile,"_pkg is new ");

```

```

put(outfile,withname.name(1..withname.len));
put(outfile," ( ");
genpar_ptr:=gen_par_ptr;
while genpar_ptr /= null loop
  Put(outfile,genpar_ptr.val);
  if genpar_ptr.link /= null then
    Put(outfile,", ");
  end if;
  genpar_ptr:=genpar_ptr.link;
end loop;
put(outfile," );");
new_line(outfile,2);

--now rename

put(outfile," procedure ");
put(outfile,queryname.name(1..queryname.len));

if query_has_in_param then
  put(outfile," (");
  inpar_ptr:= query_in_par_ptr;
  inparv_ptr:= query_in_par_val_ptr;
  while inpar_ptr /= null loop
    Put(outfile,inpar_ptr.name(1..inpar_ptr.len) );
    Put(outfile," : in ");
    inpar_ptr:=inpar_ptr.link;

    genpar_ptr:=gen_par_ptr;
    while genpar_ptr /= null loop
      if inparv_ptr.name = genpar_ptr.param then
        Put(outfile,genpar_ptr.val);
        gen_found:=True;
        exit;
      end if;
      genpar_ptr:=genpar_ptr.link;
    end loop;
  end loop;
  if gen_found=False then
    Put(outfile,inparv_ptr.name(1..inparv_ptr.len) );
  end if;
  gen_found:=False;

  inparv_ptr:= inparv_ptr.link;
  if inpar_ptr /= null then
    Put(outfile,";");
    new_line(outfile);
    set_col(outfile,13);
  else
    if query_has_out_param then
      Put(outfile,";");
      new_line(outfile);
      set_col(outfile,13);
    end if;
  end if;
end if;

```

```

        exit;
    else
        Put(outfile," ");
        new_line(outfile);
    end if;
end if;
end loop;
end if;

if query_has_out_param then
    if query_has_in_param then
        null;
    else
        put(outfile,"(");
    end if;
    outpar_ptr:= query_out_par_ptr;
    outparv_ptr:= query_out_par_val_ptr;
    while outpar_ptr /= null loop
        Put(outfile,outpar_ptr.name(1..outpar_ptr.len) );
        outpar_ptr:= outpar_ptr.link;
        Put(outfile," : out ");

        genpar_ptr:=gen_par_ptr;
        while genpar_ptr /= null loop
            if outparv_ptr.name = genpar_ptr.param then
                Put(outfile,genpar_ptr.val);
                gen_found:=True;
                exit;
            end if;
            genpar_ptr:=genpar_ptr.link;
        end loop;
        if gen_found=False then
            Put(outfile,outparv_ptr.name(1..outparv_ptr.len) );
        end if;
        gen_found:=False;

        outparv_ptr:=outparv_ptr.link;
        if outpar_ptr /= null then
            Put(outfile,";");
            new_line(outfile);
            set_col(outfile,13);
        else
            Put(outfile," ");
            new_line(outfile);
        end if;
    end loop;
end if;

new_line(outfile);
put(outfile," renames tmp_");
put(outfile,queryname.name(1..queryname.len));

```

```

    put(outfile, "_pkg.");
    put(outfile, withname.name(1..(withname.len-3)));
    put(outfile, ";");
    new_line(outfile, 2);
    put(outfile, "end ");
    put(outfile, queryname.name(1..queryname.len));
    put(outfile, "_pkg.");
    close(outfile);
end generic_operator;

```

```

procedure non_generic_adt is
begin
    create(outfile, mode=>out_file, name=>"outfile");
    create(genfile, mode=>out_file, name=>"genfile");
    put(genfile, "      ");
    put(outfile, "with ");
    put(outfile, withname.name(1..withname.len));
    put(outfile, ";");
    new_line(outfile, 2);
    put(outfile, "package ");
    put(outfile, queryname.name(1..queryname.len));
    put(outfile, "_pkg is");
    new_line(outfile, 2);

    adtop_ptr:=query_adt_op_ptr;
    r_adtop_ptr:= adt_op_ptr;
    while adtop_ptr /= null loop
        put(outfile, "  procedure ");
        Put(outfile, adtop_ptr.op_name);

        if adtop_ptr.has_inputs then
            put(outfile, " (");
            opin_par:=adtop_ptr.op_in_par;
            opin_parv:=adtop_ptr.op_in_par_val;
            while opin_par /= null loop
                put(outfile, opin_par.name);
            opin_par:=opin_par.link;
            Put(outfile, " : in ");
            put(outfile, opin_parv.name);
            opin_parv:=opin_parv.link;
            if opin_par /= null then
                Put(outfile, ";");
                new_line(outfile);
                set_col(outfile, 13);
            else
                if adtop_ptr.has_outputs then
                    Put(outfile, ";");
                    new_line(outfile);

```

```

        set_col(outfile,13);
        exit;
    else
        Put(outfile," ");
        new_line(outfile);
    end if;
end if;
end loop;
end if;

if adtop_ptr.has_outputs then
    if adtop_ptr.has_inputs then
        null;
    else
        put(outfile,"(");
    end if;
    opout_par:=adtop_ptr.op_out_par;
    opout_parv:=adtop_ptr.op_out_par_val;
    while opout_par /= null loop
        Put(outfile,opout_par.name);
        opout_par:=opout_par.link;
        Put(outfile," : out ");
        put(outfile,opout_parv.name);
        opout_parv:=opout_parv.link;
        if opout_par /= null then
            Put(outfile,";");
            new_line(outfile);
            set_col(outfile,13);
        else
            Put(outfile," )");
            new_line(outfile);
        end if;
    end loop;
end if;
new_line(outfile);
put(outfile," renames ");
put(outfile,withname.name);
put(outfile,".");
put(outfile,r_adtop_ptr.op_name);
put(outfile,";");
new_line(outfile,2);
r_adtop_ptr:=r_adtop_ptr.link;
adtop_ptr:= adtop_ptr.link;
end loop;

put(outfile,"end ");
put(outfile,queryname.name(1..queryname.len));
put(outfile,"_pkg;");
close(outfile);
end non_generic_adt;

```

```

procedure generic_adt is

begin
  create(outfile,mode=>out_file,name=>"outfile");
  create(genfile,mode=>out_file,name=>"genfile");
  put(genfile,"generic");
  put(outfile,"with ");
  put(outfile,withname.name(1..withname.len));
  put(outfile,"");
  new_line(outfile,2);
  put(outfile,"package ");
  put(outfile,queryname.name(1..queryname.len));
  put(outfile,"_pkg is");
  new_line(outfile,2);

  put(outfile," package tmp_");
  put(outfile,queryname.name(1..queryname.len));
  put(outfile,"_pkg is new ");
  put(outfile,withname.name(1..withname.len));
  put(outfile," ( ");
  genpar_ptr:=gen_par_ptr;
  while genpar_ptr /= null loop
    Put(outfile,genpar_ptr.val);
    if genpar_ptr.link /= null then
      Put(outfile,", ");
    end if;
    genpar_ptr:=genpar_ptr.link;
  end loop;
  put(outfile," );");
  new_line(outfile);

  --now rename

  adtop_ptr:=query_adt_op_ptr;
  r_adtop_ptr:= adt_op_ptr;
  while adtop_ptr /= null loop
    put(outfile," procedure ");
    Put(outfile,adtop_ptr.op_name);

    if adtop_ptr.has_inputs then
      put(outfile," (");
      opin_par:=adtop_ptr.op_in_par;
      opin_parv:=adtop_ptr.op_in_par_val;
      while opin_par /= null loop
        put(outfile,opin_par.name);
        opin_par:=opin_par.link;
        Put(outfile," : in ");
        --generic par check is to go here for generic proc
        genpar_ptr:=gen_par_ptr;
        while genpar_ptr /= null loop
          if opin_parv.name=genpar_ptr.param then

```



```

        Put(outfile,genpar_ptr.val);
        gen_found:=True;
        exit;
    end if;
    genpar_ptr:=genpar_ptr.link;
end loop;
if gen_found=False then
    put(outfile,opin_parv.name);
end if;
gen_found:=False;

opin_parv:=opin_parv.link;
if opin_par /= null then
    Put(outfile,";");
    new_line(outfile);
    set_col(outfile,13);
else
    if adtop_ptr.has_outputs then
        Put(outfile,";");
        new_line(outfile);
        set_col(outfile,13);
        exit;
    else
        Put(outfile," ");
        new_line(outfile);
    end if;
end if;
end loop;
end if;

if adtop_ptr.has_outputs then
    if adtop_ptr.has_inputs then
        null;
    else
        put(outfile,"(");
    end if;
    opout_par:=adtop_ptr.op_out_par;
    opout_parv:=adtop_ptr.op_out_par_val;
    while opout_par /= null loop
        Put(outfile,opout_par.name);
        opout_par:=opout_par.link;
        Put(outfile," : out ");
        --generic par check
        genpar_ptr:=gen_par_ptr;
        while genpar_ptr /= null loop
            if opout_parv.name=genpar_ptr.param then
                Put(outfile,genpar_ptr.val);
                gen_found:=True;
                exit;
            end if;
            genpar_ptr:=genpar_ptr.link;

```

```

        end loop;
        if gen_found=False then
            put(outfile,opout_parv.name);
        end if;
        gen_found:=False;

        opout_parv:=opout_parv.link;
        if opout_par /= null then
            Put(outfile,";");
            new_line(outfile);
            set_col(outfile,13);
        else
            Put(outfile," ");
            new_line(outfile);
        end if;
    end loop;
end if;
new_line(outfile);
put(outfile," renames tmp_");
put(outfile,queryname.name(1..queryname.len));
put(outfile,"_pkg.");
put(outfile,r_adtop_ptr.op_name);
put(outfile,";");
new_line(outfile,2);
adtop_ptr:=adtop_ptr.link;
r_adtop_ptr:=r_adtop_ptr.link;
end loop;

put(outfile,"end ");
put(outfile,queryname.name(1..queryname.len));
put(outfile,"_pkg;");
close(outfile);

end generic_adt;

end query_compose_pkg;

```

J. QUERY_PARSE.A

```
-- *** File      : query_parse.a
-- *** Author    : Dogan Ozdemir
-- *** Date      : September 1992
-- *****

with u_env, queryparser, query_lex_io, query_lex, text_io,
     query_aux_pkg, query_compose_pkg;
with component_parser, psdl_lex_io, psdl_lex,
     component_aux_pkg;

use u_env, queryparser, text_io, query_aux_pkg,
     query_compose_pkg;
use component_parser, component_aux_pkg;

procedure main is

    last1, last2      : integer:=0;

begin
    if u_env.Argc = 3 then

        component_aux_pkg.strlen(Argv(1).S, last1);
        component_aux_pkg.strlen(Argv(2).S, last2);
        psdl_lex_io.open_input(Argv(1).S(1..last1));
        psdl_lex_io.create_output;

        --starting parse retrived psdl (Argv(1))

        component_parser.yyparse;
        psdl_lex_io.close_input;
        psdl_lex_io.close_output;
        query_lex_io.open_input(Argv(2).S(1..last2));
        query_lex_io.create_output;

        --starting parse query psdl (Argv(2))

        queryparser.yyparse;
        query_lex_io.close_input;
        query_lex_io.close_output;

        if query_is_operator then

            if is_generic then
                query_compose_pkg.generic_operator;
            elsif is_generic=False then
                query_compose_pkg.non_generic_operator;
            end if;

        end if;

    end if;

end main;
```

```
elseif query_is_adt then

    if is_generic then
        query_compose_pkg.generic_adt;
    elseif is_generic=False then
        query_compose_pkg.non_generic_adt;
    end if;

end if;

else
    put_line("Wrong number of arguments");
end if;

end main;
```

APPENDIX D - SOURCE CODE FOR GUI

A. SOFTBASE.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : softbase.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- .....
-- .
-- .      softbase                      -- Main program BODY
-- .
-- .....

--| PURPOSE:
--| This is the main program of an application generated by the TAE Plus
--| Code Generator.
--|
--| REGENERATED:
--| This file is generated only once.
--|
--| NOTES:
--| To turn this into a real application, do the following:
--|
--| 1. Each panel that has event generating parameters is encapsulated by
--| a package. Each parameter that you have defined to be "event-
--| generating", has an event handler procedure in the appropriate panel's
--| package body. The panel's package body is in a file, named by
--| concatenating the string "pan_" with the panel name (followed by
--| "_b.a"). Each handler has a name that is a concatenation of the
--| parameter name and the string "_Event"
--| Add application-dependent logic to each event handler.
--| (As generated by the Workbench, each event handler simply logs the
--| occurrence of the event.)
--|
--| 2. See the "TAE Plus Ada Programmer's Guide" for directions on how
--| to compile and link this program.
--|
--|
--| ADDITIONAL NOTES:
--| In TAE Plus version 4.1, every TAE Ada application caused the screen
--| to be cleared as part of the initialization of the tae_termio package.
--| In TAE Plus version 5.0, the TAE Ada bindings have been changed so as
--| not to clear the screen by default. Simply adding a call to
--| TAE.Tae_Termio.T_Clear will produce the same results as 4.1.
--|
with TAE;
with Text_IO;
with Global;
with U_Env;
```

```

use Text_IO,Global;

-- PROGRAMMER NOTE:
-- add one "with" for each resource file in this application
with softbase_Support;

procedure softbase is

    Wpt_Event          : TAE.Tae_Wpt.Wpt_Eventptr;
    Type_of_Wpt_Event  : Tae.Wpt_Eventtype;
    User_Context_Ptr   : TAE.Tae_Wpt.Event_Context_Ptr;
    Panel_In_Resource_File : Boolean;
    UNKNOWN_WPT_EVENT  : Exception;
    L                   : Integer:=1;

begin -- softbase

    -- permit upper/lowercase file names
    TAE.Tae_Misc.F_Force_Lower (FALSE);

    TAE.Tae_Wpt.Wpt_Init ("", Global.Default_Display_Id);

    -- initialize resource file
    -- PROGRAMMER NOTE:
    -- For each resource file in this application, calls to the appropriate
    -- Initialize_All_Panels and Create_Initial_Panels must be added.

    softbase_Support.Initialize_All_Panels("/n/gemini/work/ozdemir/tae/
work/softbase.res");

    softbase_Support.Create_Initial_Panels;

    TAE.Tae_Wpt.Wpt_NewEvent (Wpt_Event);

    --ADDED
    Global.Strlen(U_Env.Argv(1).S&" ",L);
    path(1..L):=U_Env.Argv(1).S;
    Global.Strlen(U_Env.Argv(2).S&" ",L);
    proto_prefix(1..L):=U_Env.Argv(2).S;

EVENT_LOOP:
    while not Global.Application_Done loop

        -- PROGRAMMER NOTE:
        -- use Global.Set_Application_Done in "quit" event handler to exit loop

        -- Wait for the next event
        --
        TAE.Tae_Wpt.Wpt_NextEvent
            ( Event => Wpt_Event,

```

```

    Etype => Type_of_Wpt_Event );

case Type_of_Wpt_Event is

when TAE.Tae_Wpt.WPT_PARM_EVENT =>

    -- Panel event has occurred.

    -- PROGRAMMER NOTE:

    -- Get the user context (which is stored in the panel object
    -- when Wpt_NewPanel is called).
    --
    TAE.Tae_Wpt.Wpt_Extract_Context
    ( Event      => Wpt_Event,
      User_Ptr   => User_Context_Ptr );

    -- Get the parameter name
    --
    TAE.Tae_Wpt.Wpt_Extract_Parm
    ( Event      => Wpt_Event,
      Parm       => User_Context_Ptr.Parm_Name );

    -- Get target id
    --
    TAE.Tae_Wpt.Wpt_Extract_Data
    ( Event      => Wpt_Event,
      Data       => User_Context_Ptr.Datavm_Ptr );

    -- Find Vm parm object
    --
    TAE.Tae_Vm.Vm_Find
    ( Vmid       => User_Context_Ptr.Datavm_Ptr,
      Name       => User_Context_Ptr.Parm_Name,
      Vout       => User_Context_Ptr.Parm_Ptr );

    -- Dispatch event to event handler
    --
    softbase_Support.Dispatch_Panel
    ( User_Context_Ptr      => User_Context_Ptr,
      Panel_In_Resource_File => Panel_In_Resource_File );

    if (not Panel_In_Resource_File) then

        -- PROGRAMMER NOTE:
        -- For applications with more than one resource file,
        -- add a call to Dispatch_Panel for each resource file.
        --
        Text_IO.Put_Line ("Unexpected event from wpt!");
        raise TAE.Tae_Wpt.BAD_EVENT_ID;
    end if;
end case;

```

```

end if;

when TAE.Tae_Wpt.WPT_FILE_EVENT =>

    Text_IO.Put_Line ("No EVENT_HANDLER for event from external
source.");

    -- PROGRAMMER NOTE:
    -- Add code here to handle file events.
    -- Use Wpt_AddEvent and Wpt_RemoveEvent to register and remove
    -- event sources.
    -- Use Wpt_Extract_EventSource and Wpt_Extract_EventMask to get
    -- information about the event that occurred.

when TAE.Tae_Wpt.WPT_WINDOW_EVENT =>

    null;

    -- PROGRAMMER NOTE:
    -- Add code here to handle window events.
    -- WPT_WINDOW_EVENT can be caused by windows which you directly
create
    -- with X (not TAE panels), or by user acknowledgement of a
    -- Wpt_PanelMessage (therefore no default put_line statement is
    -- generated here).
    -- You MIGHT want to use Wpt_Extract_xEvent_Type here.
    --
    -- DO NOT use Wpt_Extract_Parm_xEvent since this is not
    -- a WPT_PARM_EVENT; you may get a "storage error".

when TAE.Tae_Wpt.WPT_TIMEOUT_EVENT =>

    Text_IO.Put_Line ("STUB: Event WPT_TIMEOUT_EVENT");

    -- PROGRAMMER NOTE:
    -- Add code here to handle timeout events.
    -- Use Wpt_SetTimeOut to register timeout events.

-- These are internal TAE events. The application should never see
them.
-- when TAE.Tae_Wpt.WPT_HELP_EVENT =>
-- when TAE.Tae_Wpt.WPT_INTERRUPT_EVENT =>

when others =>

    raise UNKNOWN_WPT_EVENT;

```



```

        end case;

    end loop EVENT_LOOP;

    -- PROGRAMMER NOTE:
    -- Application has ended normally.  Add application specific code to
    -- close down your application.

exception

    when UNKNOWN_WPT_EVENT =>
        Text_IO.Put ("FATAL ERROR: Unknown Wpt_NextEvent Event Type: ");
        Text_IO.Put (TAE.Wpt_Eventtype'image(Type_of_Wpt_Event) ) ;
        Text_IO.Put_Line (" ... Forced exit.");

end softbase;

```

B. GLOBAL PACKAGE

1. GLOBAL_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : global_s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Global                      -- Package SPEC
-- *
-- *****

with X_Windows;
with Text_IO;
with TAE;
with SYSTEM;
use TAE,SYSTEM,Text_IO;

package Global is

--| PURPOSE:
--| This package is automatically "with"ed in to each panel package body.
--| You can insert global variables here.
--|
--| REGENERATED:
--| This file is generated only once.
--|

    package Taefloat_IO is new Text_IO.Float_IO (TAE.Taefloat);

    Default_Display_Id : X_Windows.Display;

    --ADDED

    library          : String (1..10):= (others=>' ');
    lib_to_delete    : String (1..10):= (others=>' ');
    path             : String(1..80):= (others=>' ');
    proto_prefix     : String(1..80):= (others=>' ');
    Query_psd1       : String(1..80):= (others=>' ');
    Directory        : String(1..80):= (others=>' ');
    kwquery_outfile  : String(1..15):="kwquery_outfile";
    query_outfile    : String(1..13):="query_outfile";
    component        : String(1..80):= (others=>' ');
    directory_array  : String (1..27):= (others=>' ');
    directory_file_name: String(1..14) := "directory_file";
    lib_vec          : s_vector(1..20):= (others=> new STRING(1..10));
    file_vec         : s_vector(1..200):= (others=> new STRING(1..80));
    Is_a_directory   : Boolean:=FALSE;
```

```

Upper_directory      : Boolean:=FALSE;
Component_add        : Boolean:=FALSE;
Component_update     : Boolean:=FALSE;
Query                : Boolean:=FALSE;
current_directory    : array(1..20) of String(1..80);
directory_file       : file_type;
cur_dir_index        : integer:=1;
lib_count            : integer:=0;
num_of_comp          : integer:=1;
com                  : constant String:="$HOME/caps/src/software_base/sb ";
parse                : constant String:="$HOME/caps/src/software_base/
integrate/";

```

```

--.....
-- .
-- .      Application_Done          -- Subprogram SPEC
--
-- .....

```

```

function Application_Done
  return Boolean;

```

```

--| PURPOSE:
--| This function returns true if a "quit" event handler has called
--| Set_Application_Done, otherwise it returns false.

```

```

-- .....
-- .
-- .      Set_Application_Done      -- Subprogram SPEC
--
-- .....

```

```

procedure Set_Application_Done;

```

```

--| PURPOSE:
--| This procedure can be used by an event handler, typically a "quit"
--| button, to signal the end of the application.

```

```

--ADDED
--.....
-- .
-- .      system_call              -- Subprogram SPEC
--
-- .....

```

```

procedure system_call(command : STRING);

```

```

--| PURPOSE:
--| This procedure is used to make unix system calls from within the
--| program.

```

```

--.....
-- .
-- .      strlen                      -- Subprogram SPEC
-- .
-- .....

procedure strlen(s: in String; n: in out Integer);

--| PURPOSE:
--| This procedure is used to get the length of strings.

--.....
-- .
-- .      list_directory              -- Subprogram SPEC
-- .
-- .....

procedure list_directory(file      :in out file_type;
                        file_name:in out string;
                        file_vec  :in out s_vector;
                        I         :in out integer);

--| PURPOSE:
--| This procedure is used to obtain the contents of unix directory
--| structures.

--.....
-- .
-- .      list_components             -- Subprogram SPEC
-- .
-- .....

procedure list_components(file      :in out file_type;
                        file_name:in out string;
                        file_vec  :in out s_vector;
                        I         :in out integer);

--| PURPOSE:
--| This procedure is used to read the component list from a text file and
--| fill them into a s_vector structure to be displayed in a TAE panel.

--.....
-- .
-- .      read_directory              -- Subprogram SPEC
-- .
-- .....

procedure read_directory(file      :in out file_type;
                        file_name:in out string;
                        dir_name  :in out string);

```

```

--| PURPOSE:
--| This procedure is used to read the name of the current directory and
--| to get the path from a text file.

.....
-- .
-- .      errorstring                      -- Subprogram SPEC
-- .
-- .....

procedure errorstring(file      :in out file_type;
                      file_name:in out string;
                      err_str  :in out string);

--| PURPOSE:
--| This procedure is used to read the error message given by the
--| software base program.

.....
-- .
-- .      parse_line                      -- Subprogram SPEC
-- .
-- .....

procedure parse_line(s: in String);

--| PURPOSE:
--| This procedure is used to determine if the selected line is a
--| directory or a file and if it is a directory it gets the identity
--| of the directory.

end Global;

```

2. GLOBAL_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : global_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir

-- *****
-- *
-- *   Global                                -- Package BODY
-- *
-- *****

package body Global is

--| REGENERATED:
--| This file is generated only once.

    Is_Application_Done : Boolean := FALSE;

-- .....
-- .
-- .   Application_Done                    -- Subprogram BODY
-- .
-- .....

function Application_Done return Boolean is

begin

    return Is_Application_Done;

end Application_Done;

-- .....
-- .
-- .   Set_Application_Done                -- Subprogram BODY
-- .
-- .....

procedure Set_Application_Done is

begin

    Is_Application_Done := TRUE;

end Set_Application_Done;

--ADDED
```

```

-- .....
-- .
-- .      system_call                      -- Subprogram BODY
-- .
-- .....

procedure system_call(command : STRING) is

    procedure system_c (command : ADDRESS);
    pragma INTERFACE(C, SYSTEM_C);
    pragma INTERFACE_NAME(SYSTEM_C, "_system");
    TEMP : constant STRING := command&ASCII.NUL;
    ERROR : INTEGER;

begin
    SYSTEM_C(TEMP'ADDRESS);
end system_call;

-- .....
-- .
-- .      strlen                          -- Subprogram BODY
-- .
-- .....

procedure strlen(s: in String; n: in out Integer) is

    I : Integer := 1;

begin
    loop
        if s(I) = ' ' then
            exit;
        end if;
        I:=I+1;
    end loop;
    n:= I-1;
end strlen;

-- .....
-- .
-- .      list_directory                  -- Subprogram BODY
-- .
-- .....

procedure list_directory(file:in out file_type;
                        file_name:in out string;
                        file_vec : in out s_vector;
                        I          :in out integer)is
    len : integer :=1;

begin

```

```

    I:=1;
    open(file,mode=>in_file,name=>file_name);
    file_vec(I).all(1..2):="..";

    for cl in 3..80 loop
        file_vec(I).all(cl):=' ';
    end loop;

    while not end_of_file (file) loop
        I:=I+1;
        Text_IO.get_line(file,file_vec(I).all,len);
        for clean in (len+1)..80 loop
            file_vec(I).all(clean):=' ';
        end loop;
    end loop;
    close(file);

    exception
        when END_ERROR => null;

end list_directory;

-- .....
-- .
-- .    list_components
-- .
-- .....

procedure list_components(file      :in out file_type;
                           file_name:in out string;
                           file_vec : in out s_vector;
                           I         :in out integer)is
    len: integer :=1;

begin
    I:=1;
    open(file,mode=>in_file,name=>file_name);

    while not end_of_file (file) loop
        Text_IO.get_line(file,file_vec(I).all,len);
        for clean in (len+1)..80 loop
            file_vec(I).all(clean):=' ';
        end loop;
        I:=I+1;
    end loop;
    I:=I-1;
    close(file);
end list_components;

```



```

-- .....
-- .
-- .   read_directory
-- .
-- .....

procedure read_directory(file      :in out file_type;
                        file_name:in out string;
                        dir_name :in out string) is
    len: integer :=1;

begin
    open(file,mode=>in_file,name=>file_name);
    Text_IO.get_line(file,dir_name,len);
    close(file);
end read_directory;

-- .....
-- .
-- .   errorstring
-- .
-- .....

procedure errorstring(file      :in out file_type;
                    file_name:in out string;
                    err_str  :in out string)is
    len: integer :=1;

begin
    open(file,mode=>in_file,name=>file_name);
    Text_IO.get_line(file,err_str,len);
    if len=0 then
        Text_IO.get_line(file,err_str,len);
    end if;
    close(file);
    exception
        when END_ERROR => close(file);
end errorstring;

-- .....
-- .
-- .   parse_line
-- .
-- .....

procedure parse_line(s: in string) is

    char:character;
    N    :integer:=1;

begin

```

```

        strlen(s,N);
    if s(N)="/" then
        Is_a_directory:=TRUE;
    end if;
    if N=2 then
        if s(1..2)=".." then
            Upper_directory:=TRUE;
        end if;
    end if;
end parse_line;

end Global;

```

C. PANEL_LIBRARY PACKAGE

1. PAN_LIBRARY_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_library_s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_library                      -- Package SPEC
-- *
-- *****
```

with TAE;
with X_Windows;

package Panel_library is

```
--| PURPOSE:
--| This package encapsulates the TAE Plus panel:  library
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching.  For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   library
--|
```

Info : TAE.Tae_Wpt.Event_Context_Ptr; -- panel information

```
-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....
```

```
procedure Initialize_Panel
( Collection_Read                      -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr );  -- resource file
```

```
--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
```

```

--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read
--|
--| .....
--| .
--| .   Create_Panel                               -- Subprogram SPEC
--| .
--| .....

procedure Create_Panel
( Panel_State                               -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                          -- Panel origin is offset from
  : in X_Windows.Window                   -- this X Window.  Null_Window
  := X_Windows.Null_Window );             -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created
--|
--| .....
--| .
--| .   Connect_Panel                               -- Subprogram SPEC
--| .
--| .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                          -- Panel origin is offset from
  : in X_Windows.Window                   -- this X Window.  Null_Window
  := X_Windows.Null_Window );             -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|

```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
--| id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not be referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                               -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
  ( User_Context_Ptr           -- Wpt Event Context for a PARM
    : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_library;

```

2. PAN_LIBRARY_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_library_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_library                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Text_IO, Global;

-- One "with" statement for each connected panel.
with Panel_lbselect;
with Panel_libadd;
with Panel_lbdelete;

package body Panel_library is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below. Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler. (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   library
--| The following WorkBench operations will also cause regeneration:
--|   An item is deleted
--|   A new item is added to this panel
--|   An item's name is changed (not title)
--|   An item's data type is changed
--|   An item's generates events flag is changed
--|   An item's valids changed (if item is type string and connected)
--|   An item's connection information changed
--| For the panel items:
--|   libfunc,          help,          quit
```

```

-- .....
-- .
-- .   Initialize_Panel           -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
  ( Collection_Read
    : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "library_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "library_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_library.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_library.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel             -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window
    : in X_Windows.Window
    := X_Windows.Null_Window ) is

begin -- Create_Panel

  if Info.Panel_Id = Tae.Null_Panel_Id then
    TAE.Tae_Wpt.Wpt_NewPanel

```

```

        ( Dummy
          Data_Vm
          View_Vm
          Relative_Window
          User_Context
          Flags
          Panel_Id
        => "",
        => Info.Target,
        => Info.View,
        => Relative_Window,
        => Info,
        => Panel_State,
        => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (library) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_library.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_library.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                                -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
        ( Relative_Window
          Panel_State
        => Relative_Window,
        => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;

```



```

exception
  when TAE.Tae_Wpt.BAD_STATE =>
    Text_IO.Put_Line ("Panel_library.Connect_Panel: "
      & "Invalid panel state.");
    raise;

end Connect_Panel;

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is
begin -- Destroy_Panel

  TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

  when TAE.Tae_Wpt.BAD_PANEL_ID =>
    Text_IO.Put_Line ("Panel_library.Destroy_Panel: "
      & "Info.Panel_Id is an invalid id.");
    raise;

  when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
    -- This panel has not been created yet, or has already been destroyed.
    -- Trap this exception and do nothing.
    null;

end Destroy_Panel;

-- ++++++
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .   libfunc_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure libfunc_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.
  Value      : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count      : TAE.Taeint;

```

```

    id_file : file_type;
    id_string: String (1..80):= (others=>' ');
    id      : String (1..10):= (others=>' ');
    len,I,J : integer :=1;

begin -- libfunc_Event
    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    system_call("whoami > id_file");
    open(id_file,mode=>in_file,name=>"id_file");
    get_line(id_file,id_string,len);
    while id_string(I) /= ' ' loop
        id(J):=id_string(I);
        I:= I+1;
        J:=J+1;
    end loop;
    -- End default generated code
    -- Begin generated code for Connection
    --
    if TAE.Tae_Misc.s_equal (Value(1), "Library Selection") then null;
        Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
        Panel_lbselect.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);

    elsif TAE.Tae_Misc.s_equal (Value(1), "Library Addition") then
        if id(1..4) = "root" then
            Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
            Panel_libadd.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
        else
            TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"AUTHORIZED
PERSONNEL ONLY!");
        end if;

    elsif TAE.Tae_Misc.s_equal (Value(1), "Library Deletion") then
        if id(1..4) = "root" then
            Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
            Panel_lbdelete.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
        else
            TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"AUTHORIZED
PERSONNEL ONLY!");
        end if;
    end if;
    I:=1; J:=1;
    id:=" ";
    system_call("rm id_file");
    -- End generated code for Connection

end libfunc_Event;

```

```

-- .....
-- .
-- .    quit_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure quit_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- quit_Event
  -- Begin default generated code
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  -- End default generated code
  --ADDED
  Global.Set_Application_Done;

end quit_Event;

-- end EVENT HANDLERS
--
-----
-- .....
-- .
-- .    Dispatch_Item                  -- Subprogram BODY
-- .
--
.....

procedure Dispatch_Item
  ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

  if TAE.Tae_Misc.s_equal ("libfunc", User_Context_Ptr.Parm_Name) then
    libfunc_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("quit", User_Context_Ptr.Parm_Name) then
    quit_Event (User_Context_Ptr);
  end if;

end Dispatch_Item;

end Panel_library;

```

D. PANEL_LIBADD PACKAGE

1. PAN_LIBADD_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_libadd.s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_libadd                      -- Package SPEC
-- *
-- *****
```

with TAE;
with X_Windows;

package Panel_libadd is

```
--| PURPOSE:
--| This package encapsulates the TAE Plus panel: libadd
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   libadd
```

Info : TAE.Tae_Wpt.Event_Context_Ptr; -- panel information

```
-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....
```

```
procedure Initialize_Panel
( Collection_Read                      -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr ); -- resource file
```

```
--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

--| .....
--| .
--| .   Create_Panel                      -- Subprogram SPEC
--| .
--| .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

--| .....
--| .
--| .   Connect_Panel                     -- Subprogram SPEC
--| .
--| .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--| not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--| created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--| Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
--| id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                               -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr           -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_libadd;

```

2. PAN_LIBADD_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_libadd_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_libadd                      -- Package BODY
-- *
-- *****

with TAE; use TAE;
with Text_IO; use Text_IO;
with Global;
with Panel_lbselect;
with Panel_lbdelete;
use Global;

-- One "with" statement for each connected panel.
with Panel_library;

package body Panel_libadd is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   libadd
--|
--| The following WorkBench operations will also cause regeneration:
--|   An item is deleted
--|   A new item is added to this panel
--|   An item's name is changed (not title)
--|   An item's data type is changed
--|   An item's generates events flag is changed
--|   An item's valids changed (if item is type string and connected)
--|   An item's connection information changed
```

```

--| For the panel items:
--|   libname,           rulefile,           ok,           cancel,

--ADDED
rulefile   : String (1..15):= (others=>' ');
libr       : String (1..10):= (others=>' ');
err_string : String (1..80):= (others=>' ');
error_file : file_type;
error_fname: String(1..9) := "errorfile";
package integer_inout is new Text_IO.integer_io(integer);
use integer_inout;

-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
  ( Collection_Read
    : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "libadd_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "libadd_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_libadd.Initialize_Panel:  "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_libadd.Initialize_Panel:  "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
  ( Panel_State

```



```

        : in TAE.Tae_Wpt.Wpt_Flags
        := TAE.Tae_Wpt.WPT_PREFERRED;

Relative_Window
: in X_Windows.Window
:= X_Windows.Null_Window ) is

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
            ( Dummy          => "",
              Data_Vm        => Info.Target,
              View_Vm        => Info.View,
              Relative_Window => Relative_Window,
              User_Context    => Info,
              Flags           => Panel_State,
              Panel_Id        => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (libadd) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_libadd.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_libadd.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

```

```

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
            ( Relative_Window      => Relative_Window,
              Panel_State          => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_libadd.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .    Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_libadd.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

--+++++
--
-- begin EVENT HANDLERS
-- .....
-- .
-- .    libname_Event                      -- Subprogram SPEC & BODY

```

```

-- .
-- .....

procedure libname_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- libname_Event
  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  else
    null;
  end if;
  --ADDED
  libr(1..10):=Value(1)(1..10);
  -- End default generated code

end libname_Event;

-- .....
-- .
-- .      rulefile_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure rulefile_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- rulefile_Event
  -- Begin default generated code
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    --ADDED
    rulefile(1..15) := Value(1)(1..15);
  else
    null;
  end if;
end rulefile_Event;

```

```

        end if;
        -- End default generated code
    end rulefile_Event;

-- .....
-- .
-- .      ok_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure ok_Event
    ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;
    --ADDED
    dir   : constant String := "$HOME/caps/src/software_base/";
    N,M   : Integer:=1;
    Dummy : Boolean;
    Libcontent : file_type;
    len,S:integer:=1;

begin -- ok_Event
    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else
        null;
    end if;
    --ADDED

    open(Libcontent,mode=>in_file,name=>"libcontent");
    get(Libcontent,lib_count,1);
    if lib_count >= 10 then
        TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"ONLY 20 LIBRARY IS
PERMITTED");
    else
        global.strlen(libr,N);
        global.strlen(rulefile,M);
        global.system_call(com&"ml "&libr(1..N)&" "&dir&rulefile(1..M)&" >
"&error_fname);
        global.errorstring(error_file,error_fname,err_string);
        global.strlen(err_string,S);
        if S>1 then
            TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,err_string);
            S:=1;

```

```

        global.system_call("rm "&error_fname);
        for err in 1..80 loop
            err_string(err):=' ';
        end loop;
    else
        lib_count:=lib_count+1;
        skip_line(Libcontent);
        for I in 1..(lib_count-1) loop
            get_line(Libcontent,lib_vec(I).all,len);
            skip_line(Libcontent);
        end loop;
        lib_vec(lib_count).all:=libr;

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_lbselect.Info.Panel_Id,"selec
tion",taeint(lib_count),lib_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_lbdelete.Info.Panel_Id,"selec
tion",taeint(lib_count),lib_vec);

        Dummy:=TAE.Tae_Wpt.Wpt_Pending;--Now!
        reset(Libcontent,out_file);
        put(Libcontent,lib_count,1);
        new_line(Libcontent);
        for J in 1..lib_count loop
            put_line(Libcontent,lib_vec(J).all);
        end loop;
    end if;
end if;
close(Libcontent);
-- End default generated code
-- Begin generated code for Connection
Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
Panel_library.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection
--ADDED
TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);

end ok_Event;

-- .....
-- .
-- .      cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

```

```

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;

begin -- cancel_Event
  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  -- End default generated code
  -- Begin generated code for Connection
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
  Panel_library.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
  -- End generated code for Connection
  --ADDED
  TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);

end cancel_Event;

-- end EVENT HANDLERS
--
-----
-- .....
-- .
-- .    Dispatch_Item                      -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
  ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

  if TAE.Tae_Misc.s_equal ("libname", User_Context_Ptr.Parm_Name) then
    libname_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("rulefile", User_Context_Ptr.Parm_Name)
then
    rulefile_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("ok", User_Context_Ptr.Parm_Name) then
    ok_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
    cancel_Event (User_Context_Ptr);
  end if;

end Dispatch_Item;

end Panel_libadd;

```

E. PANEL_LBDELETE PACKAGE

1. PAN_LBDELETE_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_lbdelete_s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_lbdelete                      -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_lbdelete is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: lbdelete
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   lbdelete
--|

    Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information
    ..
    ..
    ..      Initialize_Panel                      -- Subprogram SPEC
    ..
    ..

procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr );  -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                     -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```



```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state

--| .....
--| .
--| .   Destroy_Panel                               -- Subprogram SPEC
--| .
--| .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

--| .....
--| .
--| .   Dispatch_Item                               -- Subprogram SPEC
--| .
--| .....

procedure Dispatch_Item
  ( User_Context_Ptr           -- Wpt Event Context for a PARM
    : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_lbdelete;

```

2. PAN_LBDELETE_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_lbdelete_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_lbdelete                      -- Package BODY
-- *
-- *****

with TAE; use TAE;
with Text_IO;
with Global;
use Global,Text_IO;

-- One "with" statement for each connected panel.
with Panel_library;
with Panel_ldelwarn;

package body Panel_lbdelete is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below. Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler. (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     lbdelete
--|
--| The following WorkBench operations will also cause regeneration:
--|     An item is deleted
--|     A new item is added to this panel
--|     An item's name is changed (not title)
--|     An item's data type is changed
--|     An item's generates events flag is changed
--|     An item's valids changed (if item is type string and connected)
--|     An item's connection information changed
--| For the panel items:
--|     selection,          cancel,          ok
```

```

--ADDED
package integer_inout is new Text_IO.integer_io(integer);
use integer_inout;

-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
  ( Collection_Read
    : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "lbdelete_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "lbdelete_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_lbdelete.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_lbdelete.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel                          -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window
    : in X_Windows.Window
    := X_Windows.Null_Window ) is

```

```

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
            ( Dummy          => "",
              Data_Vm        => Info.Target,
              View_Vm        => Info.View,
              Relative_Window => Relative_Window,
              User_Context    => Info,
              Flags           => Panel_State,
              Panel_Id        => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (lbdelete) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_lbdelete.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_lbdelete.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
    ( Panel_State
      : in TAE.Tae_Wpt.Wpt_Flags
      := TAE.Tae_Wpt.WPT_PREFERRED;

      Relative_Window
      : in X_Windows.Window
      := X_Windows.Null_Window ) is

--ADDED
Libcontent : file_type;
len        : integer:=1;
Dummy      : Boolean;

begin -- Connect_Panel

```

```

if Info.Panel_Id = Tae.Null_Panel_Id then
    Create_Panel
        ( Relative_Window      => Relative_Window,
          Panel_State           => Panel_State );
else
    TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    --ADDED
    open(Libcontent,mode=>in_file,name=>"libcontent");
    get(Libcontent,lib_count,1);
    skip_line(Libcontent);
    for I in 1..lib_count loop
        get_line(Libcontent,lib_vec(I).all,len);
        skip_line(Libcontent);
    end loop;

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selection",taeint(lib_count),lib_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
end if;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_lbdelete.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .    Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_lbdelete.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.

```

```

        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

-----
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .      selection_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure selection_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.
--|

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

begin -- selection_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else
        null;
    end if;
    --ADDED
    lib_to_delete(1..10):=Value(1)(1..10);
    --
    -- End default generated code

end selection_Event;

-- .....
-- .
-- .      cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

```

```

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAF.Taeint;

begin -- cancel_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else
        null;
    end if;
    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_library.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    --
    -- End generated code for Connection
    --ADDED
    TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);

end cancel_Event;

-- .....
-- .
-- .      ok_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure ok_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;

begin -- ok_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then

```

```

        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else
        null;
    end if;

    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_lidelwarn.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    --
    -- End generated code for Connection
    --ADDED
    TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);

end ok_Event;

--
--end EVENT HANDLERS
--
-----
-- .....
-- .
-- .      Dispatch_Item                      -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
  ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

    if TAE.Tae_Misc.s_equal ("selection", User_Context_Ptr.Parm_Name) then
        selection_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("ok", User_Context_Ptr.Parm_Name) then
        ok_Event (User_Context_Ptr);
    end if;

end Dispatch_Item;

end Panel_ldelete;

```


F. PANEL_LDELWARN PACKAGE

1. PAN_LDELWARN_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_ldelwarn_s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_ldelwarn                      -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_ldelwarn is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: ldelwarn
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   ldelwarn

    Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information

    .....
    ..
    ..   Initialize_Panel                  -- Subprogram SPEC
    ..
    .....

procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr ); -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                               -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                          -- Panel origin is offset from
  : in X_Windows.Window                   -- this X Window.  Null_Window
  := X_Windows.Null_Window );            -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                          -- Panel origin is offset from
  : in X_Windows.Window                   -- this X Window.  Null_Window
  := X_Windows.Null_Window );            -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--| not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--| created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--| Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
--| id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not be referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                      -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr                      -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_ldelwarn;

```

2. PAN_LDELWARN_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_ldelwarn_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_ldelwarn                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Global, Text_IO;

-- One "with" statement for each connected panel.
with Panel_library;
with Panel_lbselect;
with Panel_lbdelete;

package body Panel_ldelwarn is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     ldelwarn
--|
--| The following WorkBench operations will also cause regeneration:
--|     An item is deleted
--|     A new item is added to this panel
--|     An item's name is changed (not title)
--|     An item's data type is changed
--|     An item's generates events flag is changed
--|     An item's valids changed (if item is type string and connected)
--|     An item's connection information changed
--| For the panel items:
```

```

--|   ok,               cancel

--ADDED
package integer_inout is new Text_IO.integer_io(integer);
use integer_inout;

-- .....
-- .
-- .   Initialize_Panel           -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
  ( Collection_Read
    : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "ldelwarn_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "ldelwarn_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_ldelwarn.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_ldelwarn.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel             -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window
    : in X_Windows.Window

```

```

:= X_Windows.Null_Window ) is

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
        ( Dummy              => "",
          Data_Vm            => Info.Target,
          View_Vm            => Info.View,
          Relative_Window    => Relative_Window,
          User_Context       => Info,
          Flags              => Panel_State,
          Panel_Id           => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (ldelwarn) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_ldelwarn.Create_Panel:  "
          & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_ldelwarn.Create_Panel:  "
          & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel

```

```

        ( Relative_Window      => Relative_Window,
          Panel_State           => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_lidelwarn.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_lidelwarn.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception, and do nothing.
        null;

end Destroy_Panel;

-- .....
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .   ok_Event                          -- Subprogram SPEC & BODY
-- .
-- .....

```

```

procedure ok_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;
  --ADDED
  N : Integer:=1;
  Dummy:Boolean;
  Libcontent : file_type;
  len:integer:=1;

begin -- ok_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  else
    null;
  end if;
  --ADDED
  open(Libcontent,mode=>in_file,name=>"libcontent");
  get(Libcontent,lib_count,1);
  global.strlen(lib_to_delete,N);
  global.system_call(com&"dl "&lib_to_delete(1..N));
  skip_line(Libcontent);
  for I in 1..lib_count loop
    get_line(Libcontent,lib_vec(I).all,len);
    skip_line(Libcontent);
    if lib_to_delete=lib_vec(I).all then
      if I=lib_count then
        lib_vec(I).all:="          ";
      end if;
      lib_count:=lib_count-1;
      for J in I..(lib_count) loop
        get_line(Libcontent,lib_vec(J+1).all,len);
        skip_line(Libcontent);
        lib_vec(J).all:=lib_vec(J+1).all;
        lib_vec(J+1).all:="          ";
      end loop;
      exit;
    end if;
  end loop;

  TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_lbselect.Info.Panel_Id,"selection",taeint(lib_count),lib_vec);

```



```

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_lbdelete.Info.Panel_Id, "selec
tion", taeint(lib_count), lib_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;

    reset(Libcontent, out_file);
    put(Libcontent, lib_count, 1);
    new_line(Libcontent);
    for K in 1..(lib_count+1) loop
        put_line(Libcontent, lib_vec(K).all);
    end loop;
    close(Libcontent);

--
-- End default generated code

-- Begin generated code for Connection
--
Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
Panel_library.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection

end ok_Event;

-- .....
-- .
-- .    cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

begin -- cancel_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else
        null;
    end if;
    --
    -- End default generated code

```

```

-- Begin generated code for Connection
--
Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
Panel_library.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection

end cancel_Event;

--
-- end EVENT HANDLERS
--
-----

-- .....
-- .
-- .    Dispatch_Item                -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

    if TAE.Tae_Misc.s_equal ("ok", User_Context_Ptr.Parm_Name) then
        ok_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    end if;

end Dispatch_Item;

end Panel_ldelwarn;

```

G. PANEL_LBSELECT PACKAGE

1. PAN_LBSELECT_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_lbselect.s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_lbselect          -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_lbselect is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: lbselect
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   lbselect

Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information

-- .....
-- .
-- .   Initialize_Panel          -- Subprogram SPEC
-- .
-- .....

procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr ); -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                     -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                               -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr                -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_lbselect;

```

2. PAN_LBSELECT_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_lbselect_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_lbselect                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Global,Text_IO;

-- One "with" statement for each connected panel.
with Panel_library;
with Panel_mainmenu;

package body Panel_lbselect is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|      The panel's name is changed (not title)
--| For panel:
--|      lbselect
--|
--| The following WorkBench operations will also cause regeneration:
--|      An item is deleted
--|      A new item is added to this panel
--|      An item's name is changed (not title)
--|      An item's data type is changed
--|      An item's generates events flag is changed
--|      An item's valids changed (if item is type string and connected)
--|      An item's connection information changed
--| For the panel items:
--|      selection,          cancel,          ok
```

```

--ADDED
  libr :String (1..10):= (others=>' ');
  package integer_inout is new Text_IO.integer_io(integer);
  use integer_inout;

  -- .....
  -- .
  -- .      Initialize_Panel                      -- Subprogram BODY
  -- .
  -- .....

  procedure Initialize_Panel
    ( Collection_Read
      : in TAE.Tae_Co.Collection_Ptr ) is

  begin -- Initialize_Panel

    Info := new TAE.Tae_Wpt.Event_Context;
    Info.Collection := Collection_Read;
    TAE.Tae_Co.Co_Find (Info.Collection, "lbselect_v", Info.View);
    TAE.Tae_Co.Co_Find (Info.Collection, "lbselect_t", Info.Target);

  exception

    when TAE.UNINITIALIZED_PTR =>
      Text_IO.Put_Line ("Panel_lbselect.Initialize_Panel: "
        & "Collection_Read not initialized.");
      raise;

    when TAE.Tae_Co.NO_SUCH_MEMBER =>
      Text_IO.Put_Line ("Panel_lbselect.Initialize_Panel: "
        & "(View or Target) not in Collection.");
      raise;

  end Initialize_Panel;

  -- .....
  -- .
  -- .      Create_Panel                          -- Subprogram BODY
  -- .
  -- .....

  procedure Create_Panel
    ( Panel_State
      : in TAE.Tae_Wpt.Wpt_Flags
      := TAE.Tae_Wpt.WPT_PREFERRED;

      Relative_Window
      : in X_Windows.Window

```

```

        := X_Windows.Null_Window ) is

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
            ( Dummy
              Data_Vm
              View_Vm
              Relative_Window
              User_Context
              Flags
              Panel_Id
              => "",
              => Info.Target,
              => Info.View,
              => Relative_Window,
              => Info,
              => Panel_State,
              => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (lbselect) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_lbselect.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_lbselect.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

    --ADDED
    Libcontent : file_type;
    len        : integer:=1;
    Dummy      : Boolean;

```



```

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
        ( Relative_Window      => Relative_Window,
          Panel_State          => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
        --ADDED
        open(Libcontent,mode=>in_file,name=>"libcontent");
        get(Libcontent,lib_count,1);
        skip_line(Libcontent);
        for I in 1..lib_count loop
            get_line(Libcontent,lib_vec(I).all,len);
            skip_line(Libcontent);
        end loop;

        TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selection",taeint(li
        b_count),lib_vec);
        Dummy:=TAE.Tae_Wpt.Wpt_Pending;
        end if;

    exception

        when TAE.Tae_Wpt.BAD_STATE =>
            Text_IO.Put_Line ("Panel_lbselect.Connect_Panel:  "
            & "Invalid panel state.");
            raise;

end Connect_Panel;

-- .....
-- .
-- .    Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_lbselect.Destroy_Panel:  "
        & "Info.Panel_Id is an invalid id.");
        raise;

```

```

when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
  -- This panel has not been created yet, or has already been destroyed.
  -- Trap this exception and do nothing.
  null;

end Destroy_Panel;

-----
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .      selection_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure selection_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- selection_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  else
    null;
  end if;
  --ADDED
  libr(1..10):=Value(1)(1..10);
  --
  -- End default generated code

end selection_Event;

-- .....
-- .
-- .      cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event

```

```

    ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

begin -- cancel_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else
        null;
    end if;
    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_library.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    --
    -- End generated code for Connection
    --ADDED
    TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);

end cancel_Event;

-- .....
-- .
-- .    ok_Event                                -- Subprogram SPEC & BODY
-- .
-- .....

procedure ok_Event
    ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

begin -- ok_Event

    -- Begin default generated code
    --

```

```

TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
else
    null;
end if;
--ADDED
library := libr;
--
-- End default generated code

-- Begin generated code for Connection
--
Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
Panel_mainmenu.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection
--ADDED
TAE.Tae_Wpt.Wpt_PanelReset (Info.Panel_Id);
end ok_Event;

--
-- end EVENT HANDLERS
--
-----

-- .....
-- .
-- .    Dispatch_Item                -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
    ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

    if TAE.Tae_Misc.s_equal ("selection", User_Context_Ptr.Parm_Name) then
        selection_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("ok", User_Context_Ptr.Parm_Name) then
        ok_Event (User_Context_Ptr);
    end if;

end Dispatch_Item;

end Panel_lbselect;

```

H. PANEL_MAINMENU PACKAGE

1. PAN_MAINMENU_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_mainmenu_s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_mainmenu                      -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_mainmenu is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: mainmenu
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
file:
--|      The panel's name is changed (not title)
--| For panel:
--|      mainmenu

Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information

-- .....
-- .
-- .      Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr );  -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                     -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state
--|

-- .
-- .
-- .   Destroy_Panel           -- Subprogram SPEC
-- .
-- .

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .
-- .
-- .   Dispatch_Item           -- Subprogram SPEC
-- .
-- .

procedure Dispatch_Item
( User_Context_Ptr           -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_mainmenu;

```

2. PAN_MAINMENU_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_mainmenu_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_mainmenu          -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Text_IO,Global;

-- One "with" statement for each connected panel.
with Panel_lbselect;
with Panel_compsel;
with Panel_addfile;
with Panel_keyword;
with Panel_query;

package body Panel_mainmenu is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   mainmenu
--|
--| The following WorkBench operations will also cause regeneration:
--|   An item is deleted
--|   A new item is added to this panel
--|   An item's name is changed (not title)
--|   An item's data type is changed
--|   An item's generates events flag is changed
--|   An item's valids changed (if item is type string and connected)
```



```

--|      An item's connection information changed
--| For the panel items:
--|      cancel,          help,          browse,          component,
--|      query

--ADDED
operator_file  : file_type;
type_file     : file_type;
operator_list  : String(1..13):="operator_list";
type_list     : String(1..9):="type_list";
Dummy         : Boolean;

-- .....
-- .
-- .      Initialize_Panel          -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
  ( Collection_Read
    : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "mainmenu_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "mainmenu_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_mainmenu.Initialize_Panel:  "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_mainmenu.Initialize_Panel:  "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .      Create_Panel          -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
  ( Panel_State

```

```

        : in TAE.Tae_Wpt.Wpt_Flags
        := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window
    : in X_Windows.Window
    := X_Windows.Null_Window ) is

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
        ( Dummy                => "",
          Data_Vm              => Info.Target,
          View_Vm              => Info.View,
          Relative_Window      => Relative_Window,
          User_Context         => Info,
          Flags                => Panel_State,
          Panel_Id             => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (mainmenu) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_mainmenu.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_mainmenu.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

```

```

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
            ( Relative_Window      => Relative_Window,
              Panel_State          => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_mainmenu.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_mainmenu.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

--+++++
--
-- begin EVENT HANDLERS
--

```

```

-- .....
-- .
-- .   cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- cancel_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  --
  -- End default generated code

  -- Begin generated code for Connection
  --
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
  Panel_lbselect.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
  -- End generated code for Connection

end cancel_Event;

-- .....
-- .
-- .   browse_Event                      ,      -- Subprogram SPEC & BODY
-- .
-- .....

procedure browse_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;
  N : integer:=1;

begin -- browse_Event

```

```

-- Begin default generated code
TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
end if;
--
-- End default generated code
-- Begin generated code for Connection
--
    strlen(library,N);

    if TAE.Tae_Misc.s_equal (Value(1), "Types") then
        system_call(com&"tl "&library(1..N)&" "&"type_list");
        list_components(type_file,type_list,file_vec,num_of_comp);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_compsel.Info.Panel_Id,"compse
l",taeint(num_of_comp),file_vec);
        Dummy:=TAE.Tae_Wpt.Wpt_Pending;
        system_call("rm type_list");
        Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
        Panel_compsel.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);

        elsif TAE.Tae_Misc.s_equal (Value(1), "Operators") then
            system_call(com&"ol "&library(1..N)&" "&"operator_list");

list_components(operator_file,operator_list,file_vec,num_of_comp);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_compsel.Info.Panel_Id,"compse
l",taeint(num_of_comp),file_vec);
        Dummy:=TAE.Tae_Wpt.Wpt_Pending;
        system_call("rm operator_list");
        Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
        Panel_compsel.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    end if;
    -- End generated code for Connection

end browse_Event;
-- .....
-- .
-- .    component_Event                -- Subprogram SPEC & BODY
-- .
-- .....

procedure component_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;

```

```

I,C      :Integer:=1;
D        :Integer:=0;
ls_file  :file_type;
file_name:String(1..7) := "ls_file";
current  :String (1..80):= (others=>' ');
number   :Integer:=1;
Dummy    :Boolean;

begin -- component_Event
-- Begin default generated code
TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
end if;
--
-- End default generated code
-- Begin generated code for Connection
if TAE.Tae_Misc.s_equal (Value(1), "Add") then null;
system_call("pwd > directory_file");
read_directory(directory_file,directory_file_name,Directory);
strlen(Directory,D);
loop
    if Directory(I) = '/' then
        current_directory(cur_dir_index)(1..C+1):=current(1..C-
1)&"/ ";
        cur_dir_index:=cur_dir_index+1;
        C:=1;
    else
        current(C):=Directory(I);
        C:=C+1;
    end if;
    I:=I+1;
    if I=D+1 then exit;
    end if;
end loop;
current_directory(cur_dir_index)(1..C+1):=current(1..C-1)&"/ ";
if D>27 then
    directory_array := Directory((D-26)..D);
else
    directory_array := Directory(1..27);
end if;
system_call("ls -F "&Directory(1..D)&" > "&"ls_file");
list_directory(ls_file,file_name,file_vec,number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_addfile.Info.Panel_Id,"selspd
1",taeint(number),file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_addfile.Info.Panel_Id,"selspe
c",taeint(number),file_vec);

```

```

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_addfile.Info.Panel_Id,"selbody",taeint(number),file_vec);

TAE.Tae_Vm.Vm_SetString(Panel_addfile.Info.View,"psdldirectory",1,directory_array,TAE.Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Panel_addfile.Info.Panel_Id,"psdldirectory",Panel_addfile.Info.View,"psdldirectory");

TAE.Tae_Vm.Vm_SetString(Panel_addfile.Info.View,"specdirectory",1,directory_array,TAE.Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Panel_addfile.Info.Panel_Id,"specdirectory",Panel_addfile.Info.View,"specdirectory");

TAE.Tae_Vm.Vm_SetString(Panel_addfile.Info.View,"bodydirectory",1,directory_array,TAE.Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Panel_addfile.Info.Panel_Id,"bodydirectory",Panel_addfile.Info.View,"bodydirectory");
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;

    system_call("rm ls_file");
    system_call("rm directory_file");

    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_addfile.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    Component_add:=TRUE;

    elsif TAE.Tae_Misc.s_equal (Value(1), "Update") then null;
        Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
        Panel_addfile.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
        Component_update:=TRUE;

    end if;
    --
    -- End generated code for Connection

end component_Event;
-- .....
-- .
-- .      query_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure query_Event
    ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

    --| PURPOSE:
    --| EVENT HANDLER.  Insert application specific information.

```

```

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;
Dummy: Boolean;
D      : Integer:=0;
I,C    : Integer:=1;
number : integer:=1;
current:String (1..80):= (others=>' ');
ls_file:file_type;
file_name: String(1..7) := "ls_file";

begin -- query_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    if TAE.Tae_Misc.s_equal (Value(1), "Keyword") then
        Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
        Panel_keyword.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);

    elsif TAE.Tae_Misc.s_equal (Value(1), "PSDL") then
        strlen(path,D);
        Directory(1..D):= path(1..D);
        loop
            if Directory(I) = '/' then
                current_directory(cur_dir_index)(1..C+1):=current(1..C-
1)&"/ ";
                cur_dir_index:=cur_dir_index+1;
                C:=1;
            else
                current(C):=Directory(I);
                C:=C+1;
            end if;
            I:=I+1;
            if I=D+1 then exit;
            end if;
        end loop;
        current_directory(cur_dir_index)(1..C+1):=current(1..C-1)&"/ ";
        if D>27 then
            directory_array := Directory((D-26)..D);
        else
            directory_array := Directory(1..27);
        end if;
        system_call("ls -F "&Directory(1..D)&" > "&"ls_file");

```



```

list_directory(ls_file,file_name,file_vec,number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_query.Info.Panel_Id,"psdlselect",taeint(number),file_vec);

TAE.Tae_Vm.Vm_SetString(Panel_query.Info.View,"directorylabel",1,directory_array,TAE.Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Panel_query.Info.Panel_Id,"directorylabel",Panel_query.Info.View,"directorylabel");
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    system_call("rm ls_file");

    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_query.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);

end if;
--
-- End generated code for Connection

end query_Event;

-- end EVENT HANDLERS
-----

-- .....
-- .
-- .    Dispatch_Item                      -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is
begin -- Dispatch_Item
    if TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("browse", User_Context_Ptr.Parm_Name) then
        browse_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("component", User_Context_Ptr.Parm_Name)
then
        component_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("query", User_Context_Ptr.Parm_Name) then
        query_Event (User_Context_Ptr);
    end if;

end Dispatch_Item;

end Panel_mainmenu;

```

I. PANEL_ADDFILE PACKAGE

1. PAN_ADDFILE_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      :      pan_addfile_s.a
-- *** Generated :   May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_addfile                      -- Package SPEC
-- *
-- *****
```

with TAE;
with X_Windows;

package Panel_addfile is

```
--| PURPOSE:
--| This package encapsulates the TAE Plus panel:  addfile
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching.  For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     addfile
```

Info : TAE.Tae_Wpt.Event_Context_Ptr; -- panel information

```
-- .....
-- .
-- .      Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....
```

```
procedure Initialize_Panel
( Collection_Read                      -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr );  -- resource file
```

```
--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
panel
```

```

--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
  ( Panel_State                      -- Flags sent to Wpt_NewPanel.
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window                  -- Panel origin is offset from
    : in X_Windows.Window            -- this X Window.  Null_Window
    := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                     -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window                  -- Panel origin is offset from
    : in X_Windows.Window            -- this X Window.  Null_Window
    := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--| In this case, Relative_Window is ignored.

```

```

--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                               -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr                -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_addfile;

```

2. PAN_ADDFILE_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_addfile_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_addfile                      -- Package BODY
-- *
-- *****

with TAE; use TAE;
with Text_IO;
with Global;
use Global,Text_IO;

-- One "with" statement for each connected panel.
with Panel_mainmenu;

package body Panel_addfile is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|      The panel's name is changed (not title)
--| For panel:
--|      addfile
--|
--| The following WorkBench operations will also cause regeneration:
--|      An item is deleted
--|      A new item is added to this panel
--|      An item's name is changed (not title)
--|      An item's data type is changed
--|      An item's generates events flag is changed
--|      An item's valids changed (if item is type string and connected)
--|      An item's connection information changed
--| For the panel items:
--|      selpsdl,          selbody,          selspec,          cancel,
--|      ok,              inpsdl,           inspec,            inbody,
```

```

--|   psdldirectory,   specdirectory,   bodydirectory
--|

--ADDED
psdlfile       : String (1..80):= (others=>' ');
specfile       : String (1..80):= (others=>' ');
bodyfile       : String (1..80):= (others=>' ');
err_string     : String (1..80):= (others=>' ');
ls_file,error_file: file_type;
file_name      : String(1..7) := "ls_file";
error_fname    : String(1..9) := "errorfile";
number        : integer:=1;

-- .....
-- .
-- .   Initialize_Panel           -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
( Collection_Read
  : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "addfile_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "addfile_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_addfile.Initialize_Panel:  "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_addfile.Initialize_Panel:  "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel             -- Subprogram BODY
-- .
-- .....

```

```

procedure Create_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

begin -- Create_Panel

  if Info.Panel_Id = Tae.Null_Panel_Id then
    TAE.Tae_Wpt.Wpt_NewPanel
      ( Dummy              => "",
        Data_Vm            => Info.Target,
        View_Vm            => Info.View,
        Relative_Window    => Relative_Window,
        User_Context       => Info,
        Flags              => Panel_State,
        Panel_Id           => Info.Panel_Id );
  else
    Text_IO.Put_Line ("Panel (addfile) is already displayed.");
  end if;

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_addfile.Create_Panel: "
      & "Panel was not initialized prior to creation.");
    raise;

  when TAE.TAE_FAIL =>
    Text_IO.Put_Line ("Panel_addfile.Create_Panel: "
      & "Panel could not be created.");
    raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

```

```

        Relative_Window
        : in X_Windows.Window
        := X_Windows.Null_Window ) is

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
        ( Relative_Window      => Relative_Window,
          Panel_State          => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_addfile.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_addfile.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

```



```

--+++++
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .      selpsdl_Event          -- Subprogram SPEC & BODY
-- .
-- .....

procedure selpsdl_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.
  --|
  --| NOTES: This procedure reads the selected item and determines if it
is a
  --|          directory or a psdl file.

  Value   : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count   : TAE.Taeint;
  N,M,K,S : integer:=1;
  Dummy   : Boolean;
  psdl_vec:s_vector(1..1):= (others=> new STRING(1..80));

begin -- selpsdl_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  else
    null;
  end if;
  --ADDED
  psdlfile(1..80):=Value(1)(1..80);
  parse_line(psdlfile);
  strlen(psdlfile,N);
  strlen(Directory,M);

  if Is_a_directory then
    K:=N+M;
    cur_dir_index:=cur_dir_index+1;
    Current_directory(cur_dir_index)(1..N):=psdlfile(1..N);
    for node in (N+1)..80 loop
      Current_directory(cur_dir_index)(node):=' ';
    end loop;
    Directory(1..K):=Directory(1..M)&"/"&psdlfile(1..(N-1));

```

```

        if K>27 then
            directory_array := Directory((K-26)..K);
        else
            directory_array := Directory(1..27);
        end if;

TAE.Tae_Vm.Vm_SetString(Info.View, "psdldirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "psdldirectory", Info.View, "psdl
directory");

TAE.Tae_Vm.Vm_SetString(Info.View, "specdirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "specdirectory", Info.View, "spec
directory");

TAE.Tae_Vm.Vm_SetString(Info.View, "bodydirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "bodydirectory", Info.View, "body
directory");

        system_call("ls -F "&Directory(1..K)&" > "&ls_file");
        list_directory(ls_file, file_name, file_vec, number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "selpsdl", taeint(numbe
r), file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "selspec", taeint(numbe
r), file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "selbody", taeint(numbe
r), file_vec);
        Dummy:=TAE.Tae_Wpt.Wpt_Pending;
        system_call("rm ls_file");
        Is_a_directory:=FALSE;

    elsif Upper_directory then
        strlen(Current_directory(cur_dir_index), S);
        for drct in (M-S+1)..80 loop
            Directory(drct):=' ';
        end loop;

        if (M-S)>27 then
            directory_array := Directory(((M-S)-26)..(M-S));
        else
            directory_array := Directory(1..27);
        end if;

```

```

TAE.Tae_Vm.Vm_SetString(Info.View, "psdldirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "psdldirectory", Info.View, "psdl
directory");

TAE.Tae_Vm.Vm_SetString(Info.View, "specdirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "specdirectory", Info.View, "spec
directory");

TAE.Tae_Vm.Vm_SetString(Info.View, "bodydirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "bodydirectory", Info.View, "body
directory");

    cur_dir_index:=cur_dir_index-1;
    system_call("ls -F "&Directory(1..(M-S))&" > "&"ls_file");
    list_directory(ls_file, file_name, file_vec, number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "selpsdl", taeint(numb
er), file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "selspec", taeint(numb
er), file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "selbody", taeint(numb
er), file_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    system_call("rm ls_file");
    Upper_directory:=FALSE;

    else --A file name is selected
        psdl_vec(1).all:=psdlfile;

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "inpsdl", 1, psdl_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
end if;
--
-- End default generated code

end selpsdl_Event;

-- .....
-- .
-- .    selbody_Event                -- Subprogram SPEC & BODY
-- .
-- .....

```

```

procedure selbody_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.
  --|
  --| NOTES: This procedure reads the selected item and determines if it
is a
  --|          directory or a implementation body file.

  Value      : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count      : TAE.Taeint;
  N,M,K,S    : integer:=1;
  Dummy      : Boolean;
  body_vec: s_vector(1..1):= (others=> new STRING(1..80));

begin -- selbody_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  else
    null;
  end if;
  --ADDED
  bodyfile(1..80):=Value(1)(1..80);
  parse_line(bodyfile);
  strlen(bodyfile,N);
  strlen(Directory,M);

  if Is_a_directory then
    K:=N+M;
    cur_dir_index:=cur_dir_index+1;
    Current_directory(cur_dir_index)(1..N):=bodyfile(1..N);
    for node in (N+1)..80 loop
      Current_directory(cur_dir_index)(node):=' ';
    end loop;
    Directory(1..K):=Directory(1..M)&"/"&bodyfile(1..(N-1));

    if K>27 then
      directory_array := Directory((K-26)..K);
    else
      directory_array := Directory(1..27);
    end if;

  TAE.Tae_Vm.Vm_SetString(Info.View,"psdldirectory",1,directory_array,TAE.
Tae_Vm.P_UPDATE);

  TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"psdldirectory",Info.View,"psdl

```

```

directory");

TAE.Tae_Vm.Vm_SetString(Info.View, "specdirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "specdirectory", Info.View, "spec
directory");

TAE.Tae_Vm.Vm_SetString(Info.View, "bodydirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "bodydirectory", Info.View, "body
directory");

        system_call("ls -F "&Directory(1..K)&" > "&ls_file");
        list_directory(ls_file, file_name, file_vec, number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "selpsdl", taeint(numb
er), file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "selspec", taeint(numb
er), file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "selbody", taeint(numb
er), file_vec);
        Dummy:=TAE.Tae_Wpt.Wpt_Pending;
        system_call("rm ls_file");
        Is_a_directory:=FALSE;

        elsif Upper_directory then
            strlen(Current_directory(cur_dir_index), S);
            for drct in (M-S+1)..80 loop
                Directory(drct):=' ';
            end loop;

            if (M-S)>27 then
                directory_array := Directory(((M-S)-26)..(M-S));
            else
                directory_array := Directory(1..27);
            end if;

TAE.Tae_Vm.Vm_SetString(Info.View, "psdldirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "psdldirectory", Info.View, "psdl
directory");

TAE.Tae_Vm.Vm_SetString(Info.View, "specdirectory", 1, directory_array, TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "specdirectory", Info.View, "spec

```

```

directory");

TAE.Tae_Vm.Vm_SetString(Info.View,"bodydirectory",1,directory_array,TAE.
Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"bodydirectory",Info.View,"body
directory");

    cur_dir_index:=cur_dir_index-1;
    system_call("ls -F "&Directory(1..(M-S))&" > "&"ls_file");
    list_directory(ls_file,file_name,file_vec,number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selpsdl",taeint(numb
er),file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selspec",taeint(numb
er),file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selbody",taeint(numb
er),file_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    system_call("rm ls_file");
    Upper_directory:=FALSE;

    else
        body_vec(1).all:=bodyfile;

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"inbody",1,body_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    end if;
    --
    -- End default generated code

end selbody_Event;

-- .....
-- .
-- .    selspec_Event                -- Subprogram SPEC & BODY
-- .
-- .....

procedure selspec_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.
--|
--| NOTES: This procedure reads the selected item and determines if it
is a
--|          directory or a implementation spec file.

```

```

Value    : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count    : TAE.Taeint;
N,M,K,S  : integer:=1;
Dummy    : Boolean;
spec_vec: s_vector(1..1):= (others=> new STRING(1..80));

begin -- selspec_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else
        null;
    end if;
    --ADDED
    specfile:=Value(1)(1..80);
    parse_line(specfile);
    strlen(specfile,N);
    strlen(Directory,M);

    if Is_a_directory then
        K:=N+M;
        cur_dir_index:=cur_dir_index+1;
        Current_directory(cur_dir_index)(1..N):=specfile(1..N);
        for node in (N+1)..80 loop
            Current_directory(cur_dir_index)(node):=' ';
        end loop;
        Directory(1..K):=Directory(1..M)&"/"&specfile(1..(N-1));
        if K>27 then
            directory_array := Directory((K-26)..K);
        else
            directory_array := Directory(1..27);
        end if;

    TAE.Tae_Vm.Vm_SetString(Info.View,"psdldirectory",1,directory_array,TAE.
    Tae_Vm.P_UPDATE);

    TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"psdldirectory",Info.View,"psdl
    directory");

    TAE.Tae_Vm.Vm_SetString(Info.View,"specdirectory",1,directory_array,TAE.
    Tae_Vm.P_UPDATE);

    TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"specdirectory",Info.View,"spec
    directory");

    TAE.Tae_Vm.Vm_SetString(Info.View,"bodydirectory",1,directory_array,TAE.
    Tae_Vm.P_UPDATE);

```

```
TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"bodydirectory",Info.View,"body
directory");
```

```
    system_call("ls -F "&Directory(1..K)&" > "&"ls_file");
    list_directory(ls_file,file_name,file_vec,number);
```

```
TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selpsd1",taeint(numbe
er),file_vec);
```

```
TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selspec",taeint(numbe
er),file_vec);
```

```
TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selbody",taeint(numbe
er),file_vec);
```

```
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    system_call("rm ls_file");
    Is_a_directory:=FALSE;
```

```
    elsif Upper_directory then
        strlen(Current_directory(cur_dir_index),S);
        for drct in (M-S+1)..80 loop
            Directory(drct):=' ';
        end loop;
```

```
        if (M-S)>27 then
            directory_array := Directory((M-S)-26)..(M-S));
        else
            directory_array := Directory(1..27);
        end if;
```

```
TAE.Tae_Vm.Vm_SetString(Info.View,"psdldirectory",1,directory_array,TAE.
Tae_Vm.P_UPDATE);
```

```
TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"psdldirectory",Info.View,"psdl
directory");
```

```
TAE.Tae_Vm.Vm_SetString(Info.View,"specdirectory",1,directory_array,TAE.
Tae_Vm.P_UPDATE);
```

```
TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"specdirectory",Info.View,"spec
directory");
```

```
TAE.Tae_Vm.Vm_SetString(Info.View,"bodydirectory",1,directory_array,TAE.
Tae_Vm.P_UPDATE);
```

```
TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"bodydirectory",Info.View,"body
directory");
```

```
    cur_dir_index:=cur_dir_index-1;
    system_call("ls -F "&Directory(1..(M-S))&" > "&"ls_file");
    list_directory(ls_file,file_name,file_vec,number);
```



```

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selpsdl",taeint(number),file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selspec",taeint(number),file_vec);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"selbody",taeint(number),file_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    system_call("rm ls_file");
    Upper_directory:=FALSE;

    else
        spec_vec(1).all:=specfile;

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"inspec",1,spec_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    end if;

--
-- End default generated code

end selspec_Event;

-- .....
-- .
-- .    cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

begin -- cancel_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else
        null;
    end if;
    -- End default generated code

```

```

-- Begin generated code for Connection
--
Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
Panel_mainmenu.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
-- End generated code for Connection
--ADDED
TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);

for J in 1..80 loop
    Directory(J):=' ';
end loop;

for K in 1..cur_dir_index loop
    for L in 1..80 loop
        current_directory(K)(L):=' ';
    end loop;
end loop;

for A in 1..27 loop
    directory_array(A):=' ';
end loop;
cur_dir_index:=1;
end cancel_Event;

-- .....
-- .
-- .    ok_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure ok_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.
--|
--| NOTES: After all the files are selected, this procedure activates the
--|         software base and adds/updates the component.

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;
--ADDED
N,M,L,K,J,S:Integer:=1;

begin -- ok_Event

    -- Begin default generated code
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else

```

```

    null;
end if;

global.strlen(psdlfile,N);
global.strlen(specfile,M);
global.strlen(bodyfile,L);
global.strlen(Directory,K);
global.strlen(library,J);

if Component_add then
    global.system_call(com&"ca "&library(1..J)&" "&Directory(1..K)&"
"&psdlfile(1..N)&" "&Directory(1..K)&"/"&specfile(1..M)&"
"&Directory(1..K)&"/"&bodyfile(1..L)&" > "&error_fname);

    global.errorstring(error_file,error_fname,err_string);
    global.strlen(err_string,S);
    if S>1 then
        TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,err_string);
        S:=1;
    end if;
    global.system_call("rm "&error_fname);
    for ESA in 1..80 loop
        err_string(ESA):=' ';
    end loop;
    Component_add:=FALSE;

elseif Component_Update then
    global.system_call(com&"cu "&library(1..J)&" "&Directory(1..K)&"/
"&psdlfile(1..N)&" "&Directory(1..K)&"/"&specfile(1..M)&"
"&Directory(1..K)&"/"&bodyfile(1..L)&" > "&error_fname);
    global.errorstring(error_file,error_fname,err_string);
    global.strlen(err_string,S);
    if S>1 then
        TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,err_string);
        S:=1;
    end if;
    global.system_call("rm "&error_fname);
    for ESU in 1..80 loop
        err_string(ESU):=' ';
    end loop;
    Component_update:=FALSE;
end if;

-- End default generated code
-- Begin generated code for Connection
Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
Panel_mainmenu.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection

TAE.Tae_Wpt.Wpt_PanelReset (Info.Panel_Id);

```

```

    for X in 1..80 loop
        Directory(X):=' ';
    end loop;

    for Y in 1..cur_dir_index loop
        for P in 1..80 loop
            current_directory(Y)(P):=' ';
        end loop;
    end loop;
    for AA in 1..27 loop
        directory_array(AA):=' ';
    end loop;
    cur_dir_index:=1;

end ok_Event;

-- end EVENT HANDLERS
--
-----

-- .....
-- .
-- .    Dispatch_Item                      -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
    ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

    if TAE.Tae_Misc.s_equal ("selpsdl", User_Context_Ptr.Parm_Name) then
        selpsdl_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("selbody", User_Context_Ptr.Parm_Name)
then
        selbody_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("selspec", User_Context_Ptr.Parm_Name)
then
        selspec_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("ok", User_Context_Ptr.Parm_Name) then
        ok_Event (User_Context_Ptr);
    end if;

end Dispatch_Item;

end Panel_addfile;

```

J. PANEL_KEYWORD PACKAGE

1. PAN_KEYWORD_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_keyword_s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_keyword                      -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_keyword is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: keyword
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   keyword

Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information

-- .....!.....
-- .
-- .   Initialize_Panel                  -- Subprogram SPEC
-- .
-- .....

procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr ); -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
  ( Panel_State                      -- Flags sent to Wpt_NewPanel.
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window                  -- Panel origin is offset from
    : in X_Windows.Window            -- this X Window.  Null_Window
    := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                     -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window                  -- Panel origin is offset from
    : in X_Windows.Window            -- this X Window.  Null_Window
    := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--| not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--| created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--| Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                               -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr                               -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_keyword;

```

2. PAN_KEYWORD_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_keyword_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_keyword                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Text_IO,Global;

-- One "with" statement for each connected panel.
with Panel_mainmenu;
with Panel_compsel;

package body Panel_keyword is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below. Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler. (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     keyword
--|
--| The following WorkBench operations will also cause regeneration:
--|     An item is deleted
--|     A new item is added to this panel
--|     An item's name is changed (not title)
--|     An item's data type is changed
--|     An item's generates events flag is changed
--|     An item's valids changed (if item is type string and connected)
--|     An item's connection information changed
--| For the panel items:
--|     kwavail,          kwselected,          cancel,          ok,
```



```

--ADDED
kw_vec      : s_vector(1..100):= (others=> new STRING(1..80));
kwadd_vec   : s_vector(1..15):= (others=> new STRING(1..80));
kw_selected : String (1..80):= (others=>' ');
kw_list     : file_type;
file_name   : String(1..7) := "kw_list";
number,index : integer:=1;

-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
( Collection_Read
  : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "keyword_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "keyword_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_keyword.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_keyword.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel                          -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

```

```

    Relative_Window
    : in X_Windows.Window
    := X_Windows.Null_Window ) is

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
        ( Dummy          => "",
          Data_Vm        => Info.Target,
          View_Vm        => Info.View,
          Relative_Window => Relative_Window,
          User_Context   => Info,
          Flags          => Panel_State,
          Panel_Id       => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (keyword) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_keyword.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_keyword.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel          -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

```

```

--ADDED
Dummy:Boolean;
dir :constant String:="$HOME/caps/src/software_base/";
L   :integer:=1;

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
            ( Relative_Window      => Relative_Window,
              Panel_State          => Panel_State );
    end if;

    --ADDED
    strlen(library,L);
    system_call(com&"kwl "&library(1..L)&" "&"kw_list");
    list_components(kw_list,file_name,kw_vec,number);
    TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);

    TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"kwavail",taeint(number),kw_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    system_call("rm kw_list");

    exception

        when TAE.Tae_Wpt.BAD_STATE =>
            Text_IO.Put_Line ("Panel_keyword.Connect_Panel:  "
                & "Invalid panel state.");
            raise;

end Connect_Panel;

-- .....
-- .
-- .    Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

    exception

        when TAE.Tae_Wpt.BAD_PANEL_ID =>
            Text_IO.Put_Line ("Panel_keyword.Destroy_Panel:  "
                & "Info.Panel_Id is an invalid id.");

```

```

        raise;

when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
    -- This panel has not been created yet, or has already been destroyed.
    -- Trap this exception and do nothing.
    null;

end Destroy_Panel;

-----
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .      kwavail_Event          -- Subprogram SPEC & BODY
-- .
-- .....

procedure kwavail_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.
--|

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

begin -- kwavail_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --ADDED
    kw_selected(1..80):=Value(1)(1..80);
    kwadd_vec(index).all:=kw_selected;

    TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"kwselected",taeint(i
ndex),kwadd_vec);
    index:=index+1;

    --
    -- End default generated code

end kwavail_Event;

```

```

-- .....
-- .
-- .   cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- cancel_Event
  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  else
    null;
  end if;
  --
  -- End default generated code

  -- Begin generated code for Connection
  --
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
  Panel_mainmenu.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
  --
  -- End generated code for Connection

  --ADDED
  for J in 1..(index-1) loop
    for K in 1..80 loop
      kwadd_vec(J).all(K):=' ';
    end loop;
  end loop;
  index:=1;
  kw_selected(1..5) := "      ";
  TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);

end cancel_Event;

-- .....
-- .
-- .   ok_Event                          -- Subprogram SPEC & BODY
-- .
-- .....

```

```

procedure ok_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;
  --ADDED
  N      : integer:=1;
  kwquery_file: file_type;
  Dummy   : Boolean;
  kwq_outfile : file_type;

begin -- ok_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  --ADDED
  if kw_selected(1..5) = "      " then
    TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"NO KEYWORD IS
SELECTED");
  else

    create(kwquery_file,mode=>out_file,name=>"query_file");
    for I in 1..(index-1) loop
      put_line(kwquery_file,kwadd_vec(I).all);
    end loop;
    strlen(library,N);
    system_call(com&"kwq "&library(1..N)&" "&"query_file"&
"&kwquery_outfile);
    delete(kwquery_file);
    list_components(kwq_outfile,kwquery_outfile,file_vec,num_of_comp);

TAE.Tae_Wpt.Wpt_SetStringConstraints (Panel_compsel.Info.Panel_Id, "compse
l", taeint(num_of_comp), file_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    system_call("rm kwquery_outfile");

  --
  -- End default generated code

  -- Begin generated code for Connection
  --
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);

```

```

Panel_compsel.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection

--ADDED

for J in 1..(index-1) loop
  for K in 1..80 loop
    kwadd_vec(J).all(K):=' ';
  end loop;
end loop;
index:=1;
kw_selected(1..5) := "      ";
TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);

end if;

end ok_Event;

--
-- end EVENT HANDLERS
--
-----

-- .....
-- .
-- .    Dispatch_Item          -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
  ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item    ,

  if TAE.Tae_Misc.s_equal ("kwavail", User_Context_Ptr.Parm_Name) then
    kwavail_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
    cancel_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("ok", User_Context_Ptr.Parm_Name) then
    ok_Event (User_Context_Ptr);
  end if;

end Dispatch_Item;

end Panel_keyword;

```

K. PANEL_QUERY PACKAGE

1. PAN_QUERY_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_query_s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_query                      -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_query is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: query
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   query

Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information

-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr ); -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
--|
```



```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                     -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specifiec Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state

--| .....
--| .
--| .   Destroy_Panel                               -- Subprogram SPEC
--| .
--| .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

--| .....
--| .
--| .   Dispatch_Item                               -- Subprogram SPEC
--| .
--| .....

procedure Dispatch_Item
  ( User_Context_Ptr          -- Wpt Event Context for a PARM
    : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_query;

```

2. PAN_QUERY_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_query_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_query                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Global,Text_IO;

-- One "with" statement for each connected panel.
with Panel_mainmenu;
with Panel_compsel;

package body Panel_query is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below. Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler. (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|      The panel's name is changed (not title)
--| For panel:
--|      query
--|
--| The following WorkBench operations will also cause regeneration:
--|      An item is deleted
--|      A new item is added to this panel
--|      An item's name is changed (not title)
--|      An item's data type is changed
--|      An item's generates events flag is changed
--|      An item's valids changed (if item is type string and connected)
--|      An item's connection information changed
--| For the panel items:
--|      psdlselect,      psdlin,      directorylabel,  cancel,
```

```

--|   ok

--ADDED
psdlfile :String (1..80):= (others=>' ');
current  :String (1..80):= (others=>' ');
ls_file  :file_type;
q_outfile:file_type;
file_name:String(1..7) := "ls_file";
number   :integer:=1;
Dummy    :Boolean;

-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
( Collection_Read
  : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "query_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "query_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_query.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_query.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel                          -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
( Panel_State

```

```

        : in TAE.Tae_Wpt.Wpt_Flags
        := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window
    : in X_Windows.Window
    := X_Windows.Null_Window ) is

--| NOTES: (none)

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
        ( Dummy
          => "",
          Data_Vm
          => Info.Target,
          View_Vm
          => Info.View,
          Relative_Window
          => Relative_Window,
          User_Context
          => Info,
          Flags
          => Panel_State,
          Panel_Id
          => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (query) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_query.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_query.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window

```

```

        : in X_Windows.Window
        := X_Windows.Null_Window ) is

--| NOTES: (none)
begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
        ( Relative_Window      => Relative_Window,
          Panel_State          => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_query.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_query.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

--+++++
--

```

```

-- begin EVENT HANDLERS
--
-- .....
-- .
--     psdlselect_Event                -- Subprogram SPEC & BODY
-- .
-- .....

procedure psdlselect_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;
    --ADDED
    N,M,K,S : integer:=1;
    Dummy    :Boolean;
    psdl_vec:s_vector(1..1):= (others=> new STRING(1..80));

begin -- psdlselect_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --ADDED
    psdlfile:=Value(1)(1..80);
    parse_line(psdlfile);
    strlen(psdlfile,N);
    strlen(Directory,M);
    if Is_a_directory then
        K:=N+M;
        Directory(1..K):=Directory(1..M)&"/"&psdlfile(1..(N-1));

        if K>27 then
            directory_array := Directory((K-26)..K);
        else
            directory_array := Directory(1..27);
        end if;

    TAE.Tae_Vm.Vm_SetString(Info.View,"directorylabel",1,directory_array,TAE
    .Tae_Vm.P_UPDATE);

    TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"directorylabel",Info.View,"dir
    ectorylabel");

    system_call("ls -F "&Directory(1..K)&" > "&"ls_file");

```

```

list_directory(ls_file, file_name, file_vec, number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "psdlselect", taeint(n
umber), file_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    cur_dir_index:=cur_dir_index+1;
    Current_directory(cur_dir_index)(1..N):=psdlfile(1..N);
    for node in (N+1)..80 loop
        Current_directory(cur_dir_index)(node):=' ';
    end loop;
    system_call("rm ls_file");
    Is_a_directory:=FALSE;

    elsif Upper_directory then
        strlen(Current_directory(cur_dir_index), S);
        for drct in (M-S+1)..80 loop
            Directory(drct):=' ';
        end loop;
        if (M-S)>27 then
            directory_array := Directory(((M-S)-26)..(M-S));
        else
            directory_array := Directory(1..27);
        end if;

TAE.Tae_Vm.Vm_SetString(Info.View, "directorylabel", 1, directory_array, TAE
.Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id, "directorylabel", Info.View, "dir
ectorylabel");

    cur_dir_index:=cur_dir_index-1;
    system_call("ls -F "&Directory(1..(M-S))&" > "&"ls_file");
    list_directory(ls_file, file_name, file_vec, number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "psdlselect", taeint(n
umber), file_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    system_call("rm ls_file");
    Upper_directory:=FALSE;

    else --A file name is selected
        psdl_vec(1).all:=psdlfile;

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id, "psdlin", 1, psdl_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    end if;

--
-- End default generated code

end psdlselect_Event;

```



```

-- .....
-- .
-- .   cancel_Event                               -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- cancel_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  --
  -- End default generated code

  -- Begin generated code for Connection
  --
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
  Panel_mainmenu.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
  --
  -- End generated code for Connection
  --ADDED
  TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);
  for J in 1..80 loop
    Directory(J):=' ';
  end loop;

  for K in 1..cur_dir_index loop
    for L in 1..80 loop
      current_directory(K)(L):=' ';
    end loop;
  end loop;
  for A in 1..27 loop
    directory_array(A):=' ';
  end loop;
  cur_dir_index:=1;

end cancel_Event;

```

```

-- .....
-- .
-- .   ok_Event                               -- Subprogram SPEC & BODY
-- .
-- .....

procedure ok_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;
  --ADDED
  N,K,J,S:Integer:=1;
  teststring:String(1..80):= (others=>' ');

begin -- ok_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  --ADDED
  strlen(psdlfile,N);
  strlen(Directory,K);
  strlen(library,J);
  system_call(com&"cq "&library(1..J)&" "&Directory(1..K)&"/
"&psdlfile(1..N)&" "&query_outfile);
  list_components(q_outfile,query_outfile,file_vec,num_of_comp);

  --strlen(file_vec(1).all,S);
  open(q_outfile,mode=>in_file,name=>query_outfile);
  Text_IO.get_line(q_outfile,teststring,S);
  close(q_outfile);
  if teststring(1..5)="      " then
    TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"NO MATCHING COMPONENT
IS FOUND");
    system_call("rm query_outfile");
  else

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_compsel.Info.Panel_Id,"compse
1",taeint(num_of_comp),file_vec);
  Dummy:=TAE.Tae_Wpt.Wpt_Pending;
  system_call("rm query_outfile");
  Query:= True;
  Query_psdl(1..N):= psdlfile(1..N);

```

```

    system_call("cp "&Directory(1..K)&"/"&psdlfile(1..N)&" .");
-- End default generated code
-- Begin generated code for Connection
Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
Panel_compsel.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection
--ADDED
TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);
for X in 1..80 loop
    Directory(X):=' ';
end loop;

for Y in 1..cur_dir_index loop
    for Z in 1..80 loop
        current_directory(Y)(Z):=' ';
    end loop;
end loop;
for AA in 1..27 loop
    directory_array(AA):=' ';
end loop;
cur_dir_index:=1;

end if;

end ok_Event;
-- end EVENT HANDLERS
-----

-- .....
-- .
-- .    Dispatch_Item                      -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

    if TAE.Tae_Misc.s_equal ("psdlselect", User_Context_Ptr.Parm_Name)
then
        psdlselect_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("ok", User_Context_Ptr.Parm_Name) then
        ok_Event (User_Context_Ptr);
    end if;
end Dispatch_Item;

end Panel_query;

```

L. PANEL_COMPSEL PACKAGE

1. PAN_COMPSEL_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_compsel_s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_compsel                      -- Package SPEC
-- *
-- *****
```

with TAE;
with X_Windows;

package Panel_compsel is

```
--| PURPOSE:
--| This package encapsulates the TAE Plus panel:  compsel
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     compsel
--|
```

```
Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information
-- .....
-- .
-- .      Initialize_Panel                  -- Subprogram SPEC
-- .
-- .....
```

```
procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr ); -- resource file
```

```
--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
```

```

--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read
--|

--| .....
--| .
--|   Create_Panel                      -- Subprogram SPEC
--| .
--| .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created
--|

--| .....
--| .
--|   Connect_Panel                    -- Subprogram SPEC
--| .
--| .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specifiec Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.

```

```

--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state
--|
--| .....
--| .
--| .   Destroy_Panel                               -- Subprogram SPEC
--| .
--| .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.
--|
--| .....
--| .
--| .   Dispatch_Item                               -- Subprogram SPEC
--| .
--| .....

procedure Dispatch_Item
( User_Context_Ptr                -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific
--|

end Panel_compsel;

```

2. PAN_COMPSEL_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_compsel_b.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_compsel                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Global, Text_IO;

-- One "with" statement for each connected panel.
with Panel_mainmenu;
with Panel_viewpsdl;
with Panel_select;

package body Panel_compsel is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   compsel
--|
--| The following WorkBench operations will also cause regeneration:
--|   An item is deleted
--|   A new item is added to this panel
--|   An item's name is changed (not title)
--|   An item's data type is changed
--|   An item's generates events flag is changed
--|   An item's valids changed (if item is type string and connected)
--|   An item's connection information changed
--| For the panel items:
```

```

--|   compsel,           cancel,           view,           select,
--|

--ADDED
comp           : String(1..50) := (others=>' ');

-- .....
-- .
-- .   Initialize_Panel           -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
( Collection_Read
  : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "compsel_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "compsel_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_compsel.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_compsel.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel           -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window

```



```

        := X_Windows.Null_Window ) is

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
            ( Dummy          => "",
              Data_Vm        => Info.Target,
              View_Vm        => Info.View,
              Relative_Window => Relative_Window,
              User_Context    => Info,
              Flags           => Panel_State,
              Panel_Id        => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (compsel) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_compsel.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_compsel.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
    ( Panel_State
      : in TAE.Tae_Wpt.Wpt_Flags
      := TAE.Tae_Wpt.WPT_PREFERRED;

      Relative_Window
      : in X_Windows.Window
      := X_Windows.Null_Window ) is

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel

```

```

        ( Relative_Window      => Relative_Window,
          Panel_State           => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_compsel.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .   Destroy_Panel                               -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_compsel.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

--+++++
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .   compsel_Event                               -- Subprogram SPEC & BODY
-- .
-- .....

```

```

procedure compsel_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.
  --|
  --| NOTES: Selects the component from the list

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- compsel_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  else
    null;
  end if;
  --ADDED
  comp(1..80):=Value(1)(1..80);

  --
  -- End default generated code

end compsel_Event;

-- .....
-- .
-- .      cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event      ,
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.
  --|
  --| NOTES: (none)

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;
  Q      : Integer:=1;

begin -- cancel_Event

  -- Begin default generated code

```

```

--
TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
else
    null;
end if;
if Query then
    Query:=False;
    strlen(Query_psd1,Q);
    system_call("rm "&Query_psd1(1..Q));
end if;
--
-- End default generated code

-- Begin generated code for Connection
--
Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
Panel_mainmenu.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection
--ADDED
TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);
for Clean in 1..(num_of_comp) loop
    for ch in 1..80 loop
        file_vec(Clean).all(ch):=' ';
    end loop;
end loop;
comp(1..5):="      ";

end cancel_Event;

-- .....
-- .
-- .   view_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure view_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.
--|
--| NOTES: (none)

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;
L,S   : Integer:=1;

```

```

begin -- view_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --ADDED
    if comp(1..5) = "      " then
        TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"NO COMPONENT IS
SELECTED");
    else
        component := comp;
        strlen(component,L);
        strlen(library,S);

        system_call(com&"cv "&library(1..S)&" "&component(1..L)&" outpsdl
outspec outbody");
    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_viewpsdl.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    --
    -- End generated code for Connection
    end if;

end view_Event;

-- .....
-- .
-- .   select_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure select_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

    --| PURPOSE:
    --| EVENT HANDLER.  Insert application specific information.
    --|
    --| NOTES: Selected component will be further manipulated if this
procedure
    --|          is invoked.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

```

```

begin -- select_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --ADDED
    if comp(1..5) = "      " then
        TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"NO COMPONENT IS
SELECTED");
    else
        component := comp;

        -- Begin generated code for Connection
        Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
        Panel_select.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
        -- End generated code for Connection
    end if;

end select_Event;

-- end EVENT HANDLERS
-----

-- .....
-- .
-- .    Dispatch_Item                -- Subprogram BODY
-- .
-- .....

.....

procedure Dispatch_Item
( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

    if TAE.Tae_Misc.s_equal ("compsel", User_Context_Ptr.Parm_Name) then
        compsel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("view", User_Context_Ptr.Parm_Name) then
        view_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("select", User_Context_Ptr.Parm_Name) then
        select_Event (User_Context_Ptr);
    end if;

end Dispatch_Item;

end Panel_compsel;

```

M. PANEL_VIEWPSDL PACKAGE

1. PAN_VIEWPSDL_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_viewpsdl.s.a
-- *** Generated  : Jul 13 16:13:09 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_viewpsdl                      -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_viewpsdl is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: viewpsdl
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file.
--|   The panel's name is changed (not title)
--| For panel:
--|   viewpsdl

    Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information

-- .....
-- .
-- .      Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Initialize_Panel
    ( Collection_Read          -- TAE Collection read from
      ; in TAE.Tae_Co.Collection_Ptr ); -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
--   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
--   Connect_Panel                    -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```



```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                      -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr                      -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_viewsdl;

```

2. PAN_VIEWPSDL_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_viewpsdl_b.a
-- *** Generated  : Jul 13 16:13:09 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_viewpsdl                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Global, Text_IO;

-- One "with" statement for each connected panel.
with Panel_compsel;
with Panel_viewspec;

package body Panel_viewpsdl is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below. Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler. (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     viewpsdl
--|
--| The following WorkBench operations will also cause regeneration:
--|     An item is deleted
--|     A new item is added to this panel
--|     An item's name is changed (not title)
--|     An item's data type is changed
--|     An item's generates events flag is changed
--|     An item's valids changed (if item is type string and connected)
--|     An item's connection information changed
--| For the panel items:
```

```

--|   cancel,           viewspec

--   .....
--   .
--   .   Initialize_Panel           -- Subprogram BODY
--   .
--   .....

procedure Initialize_Panel
( Collection_Read
  : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "viewpsdl_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "viewpsdl_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_viewpsdl.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_viewpsdl.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

--   .....
--   .
--   .   Create_Panel           -- Subprogram BODY
--   .
--   .....

procedure Create_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

begin -- Create_Panel

```

```

if Info.Panel_Id = Tae.Null_Panel_Id then
  TAE.Tae_Wpt.Wpt_NewPanel
    ( Dummy
      Data_Vm
      View_Vm
      Relative_Window
      User_Context
      Flags
      Panel_Id
      => "",
      => Info.Target,
      => Info.View,
      => Relative_Window,
      => Info,
      => Panel_State,
      => Info.Panel_Id );
else
  Text_IO.Put_Line ("Panel (viewpsdl) is already displayed.");
end if;

exception

when TAE.UNINITIALIZED_PTR =>
  Text_IO.Put_Line ("Panel_viewpsdl.Create_Panel: "
    & "Panel was not initialized prior to creation.");
  raise;

when TAE.TAE_FAIL =>
  Text_IO.Put_Line ("Panel_viewpsdl.Create_Panel: "
    & "Panel could not be created.");
  raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window
    : in X_Windows.Window
    := X_Windows.Null_Window ) is

  Dummy: Boolean;
  L      : Integer:=1;
  label: String(1..80):=(others=>' ');

begin -- Connect_Panel

  if Info.Panel_Id = Tae.Null_Panel_Id then
    Create_Panel
      ( Relative_Window
        => Relative_Window,

```

```

        Panel_State          => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;

    --ADDED
    strlen(component,L);
    label(1..(L+5)):=component(1..L)&".psdl";

TAE.Tae_Vm.Vm_SetString(Info.View,"psdllabel",1,label(1..(L+5)),TAE.Tae_
Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"psdllabel",Info.View,"psdllabe
l");

    Dummy:=TAE.Tae_Wpt.Wpt_Pending;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_viewpsdl.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .    Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is
begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_viewpsdl.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

```

```

-----
--
-- begin EVENT HANDLERS
--
-- .....
-- .
--      cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

begin -- cancel_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --ADDED
    system_call("rm outpsdl");
    system_call("rm outspec");
    system_call("rm outbody");
    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_compsel.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    --
    -- End generated code for Connection

end cancel_Event;

-- .....
-- .
--      viewspec_Event                    -- Subprogram SPEC & BODY
-- .
-- .....

procedure viewspec_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

```

```

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.
--|
--| NOTES: (none)

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;

begin -- viewspec_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;

    -- End default generated code
    -- Begin generated code for Connection
    Panel_viewspec.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    -- End generated code for Connection

end viewspec_Event;
-- end EVENT HANDLERS
-----

-- .....
-- .
-- .    Dispatch_Item                      -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

    if TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("viewspec", User_Context_Ptr.Parm_Name)
then
        viewspec_Event (User_Context_Ptr);
    end if;

end Dispatch_Item;

end Panel_viewspsdl;

```

N. PANEL_VIEWSPEC PACKAGE

1. PAN_VIEWSPEC_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_viewspec_s.a
-- *** Generated  : Jul 13 16:13:09 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_viewspec                      -- Package SPEC
-- *
-- *****
```

```
with TAE;
with X_Windows;
```

package Panel_viewspec is

```
--| PURPOSE:
--| This package encapsulates the TAE Plus panel:  viewspec
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching.  For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     viewspec
```

```
Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information
```

```
-- .....
-- .
-- .      Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....
```

```
procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr );  -- resource file
```

```
--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
--|
```



```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                     -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--| not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--| created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--| Panel_State is an invalid state

--| .....
--| .
--| . Destroy_Panel -- Subprogram SPEC
--| .
--| .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

--| .....
--| .
--| . Dispatch_Item -- Subprogram SPEC
--| .
--| .....

procedure Dispatch_Item
( User_Context_Ptr -- Wpt Event Context for a PARM
: in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_viewspec;

```

2. PAN_VIEWSPEC_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_viewspec_b.a
-- *** Generated  : Jul 13 16:13:09 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_viewspec                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Text_IO,Global;

-- One "with" statement for each connected panel.
with Panel_viewspsdl;
with Panel_viewbody;

package body Panel_viewspec is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below. Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler. (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     viewspec
--|
--| The following WorkBench operations will also cause regeneration:
--|     An item is deleted
--|     A new item is added to this panel
--|     An item's name is changed (not title)
--|     An item's data type is changed
--|     An item's generates events flag is changed
--|     An item's valids changed (if item is type string and connected)
--|     An item's connection information changed
--| For the panel items:
--|     cancel,          viewbody
```

```

-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
  ( Collection_Read
    : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "viewspec_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "viewspec_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_viewspec.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_viewspec.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel                        -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window
    : in X_Windows.Window
    := X_Windows.Null_Window ) is

begin -- Create_Panel

  if Info.Panel_Id = Tae.Null_Panel_Id then
    TAE.Tae_Wpt.Wpt_NewPanel

```

```

        ( Dummy
          Data_Vm
          View_Vm
          Relative_Window
          User_Context
          Flags
          Panel_Id
        => "",
        => Info.Target,
        => Info.View,
        => Relative_Window,
        => Info,
        => Panel_State,
        => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (viewspec) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_viewspec.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_viewspec.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

    Dummy:Boolean;
    L : Integer:=1;
    label: String(1..80):=(others=>' ');

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
        ( Relative_Window
          Panel_State
        => Relative_Window,
        => Panel_State );
    else

```

```

    TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
end if;

--ADDED
strlen(component,L);
label(1..(L+5)):=component(1..L)&".spec";

TAE.Tae_Vm.Vm_SetString(Info.View,"speclabel",1,label(1..(L+5)),TAE.Tae_
Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"speclabel",Info.View,"speclabe
l");
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_viewspec.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .    Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is
begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_viewspec.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

--+++++
--

```

```

-- begin EVENT HANDLERS
--
-- .....
-- .
-- .      cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

begin -- cancel_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_viewsdl.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    --
    -- End generated code for Connection

end cancel_Event;

,

-- .....
-- .
-- .      viewbody_Event                    -- Subprogram SPEC & BODY
-- .
-- .....

procedure viewbody_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;

```

```

begin -- viewbody_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    Panel_viewbody.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    --
    -- End generated code for Connection

end viewbody_Event;

--
-- end EVENT HANDLERS
--
-----

-- .....
-- .
-- .    Dispatch_Item                -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
    ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item    ,

    if TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("viewbody", User_Context_Ptr.Parm_Name)
then
        viewbody_Event (User_Context_Ptr);
    end if;

    end Dispatch_Item;

end Panel_viewspec;

```


O. PANEL_VIEWBODY PACKAGE

1. PAN_VIEWBODY_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_viewbody_s.a
-- *** Generated  : Jul 13 16:13:09 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_viewbody                      -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_viewbody is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: viewbody
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     viewbody

    Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information

    .....

    .
    .      Initialize_Panel                  -- Subprogram SPEC
    .
    .....

procedure Initialize_Panel
    ( Collection_Read          -- TAE Collection read from
      : in TAE.Tae_Co.Collection_Ptr );  -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read
--
-- .....
-- .
-- .   Create_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                               -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                          -- Panel origin is offset from
  : in X_Windows.Window                   -- this X Window.  Null_Window
  := X_Windows.Null_Window );            -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created
--
-- .....
-- .
-- .   Connect_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                          -- Panel origin is offset from
  : in X_Windows.Window                   -- this X Window.  Null_Window
  := X_Windows.Null_Window );            -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--| In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--| not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--| created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--| Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                               -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr          -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_viewbody;

```

2. PAN_VIEWBODY_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_viewbody_b.a
-- *** Generated  : Jul 13 16:13:09 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_viewbody                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Text_IO,Global;

-- One "with" statement for each connected panel.
with Panel_viewspec;

package body Panel_viewbody is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   viewbody
--|
--| The following WorkBench operations will also cause regeneration:
--|   An item is deleted
--|   A new item is added to this panel
--|   An item's name is changed (not title)
--|   An item's data type is changed
--|   An item's generates events flag is changed
--|   An item's valids changed (if item is type string and connected)
--|   An item's connection information changed
--| For the panel items:
--|   cancel
```

```

-- .....
-- .
-- .   Initialize_Panel           -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
  ( Collection_Read
    : in TAE.Tae_Co.Collection_Ptr ) is

--| NOTES: (none)

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "viewbody_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "viewbody_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_viewbody.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_viewbody.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel             -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window
    : in X_Windows.Window
    := X_Windows.Null_Window ) is

begin -- Create_Panel

```

```

if Info.Panel_Id = Tae.Null_Panel_Id then
    TAE.Tae_Wpt.Wpt_NewPanel
        ( Dummy
          Data_Vm
          View_Vm
          Relative_Window
          User_Context
          Flags
          Panel_Id
          => "",
          => Info.Target,
          => Info.View,
          => Relative_Window,
          => Info,
          => Panel_State,
          => Info.Panel_Id );
else
    Text_IO.Put_Line ("Panel (viewbody) is already displayed.");
end if;

exception

when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_viewbody.Create_Panel: "
        & "Panel was not initialized prior to creation.");
    raise;

when TAE.TAE_FAIL =>
    Text_IO.Put_Line ("Panel_viewbody.Create_Panel: "
        & "Panel could not be created.");
    raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

Dummy:Boolean;
L      : Integer:=1;
label: String(1..80):=(others=>' ');

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
            ( Relative_Window          => Relative_Window,

```

```

        Panel_State          => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;
    --ADDED
    strlen(component,L);
    label(1..(L+5)):=component(1..L)&".body";

TAE.Tae_Vm.Vm_SetString(Info.View,"bodylabel",1,label(1..(L+5)),TAE.Tae_
Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"bodylabel",Info.View,"bodylabel");
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_viewbody.Connect_Panel: "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .    Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_viewbody.Destroy_Panel: "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

```

```

-- begin EVENT HANDLERS
-- .....
-- .
-- .      cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- cancel_Event
  -- Begin default generated code
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  -- End default generated code
  -- Begin generated code for Connection
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
  Panel_viewspec.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
  -- End generated code for Connection
end cancel_Event;

-- end EVENT HANDLERS
-----

-- .....
-- .
-- .      Dispatch_Item                      -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
  ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

  if TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
    cancel_Event (User_Context_Ptr);
  end if;

end Dispatch_Item;

end Panel_viewbody;

```


P. PANEL_SELECT PACKAGE

1. PAN_SELECT_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_select_s.a
-- *** Generated  : Aug 12 11:50:32 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_select                      -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_select is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: select
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   select

    Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information

    .....!.....
    .
    .   Initialize_Panel                  -- Subprogram SPEC
    .
    .....

procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr );  -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read
--|

--
-- .....
-- .
-- .   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created
--|

--
-- .....
-- .
-- .   Connect_Panel                     -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State                      ,
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|

```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                               -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr                -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_select;

```

2. PAN_SELECT_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File:          pan_select_b.a
-- *** Generated:     Aug 12 11:50:32 1992
-- *****
-- *
-- *   Panel_select          -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Global, Text_IO;

-- One "with" statement for each connected panel.
with Panel_cdelwarn;
with Panel_savecomp;
with Panel_compsel;
with Panel_integrt;

package body Panel_select is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   select
--|
--| The following WorkBench operations will also cause regeneration:
--|   An item is deleted
--|   A new item is added to this panel
--|   An item's name is changed (not title)
--|   An item's data type is changed
--|   An item's generates events flag is changed
--|   An item's valids changed (if item is type string and connected)
--|   An item's connection information changed
--| For the panel items:
```

```

--| delete,          save,          print,          integrate,
--| cancel,         help

-- .....
-- .
--   Initialize_Panel          -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
( Collection_Read
  : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "select_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "select_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_select.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_select.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
--   Create_Panel          -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

begin -- Create_Panel

```

```

if Info.Panel_Id = Tae.Null_Panel_Id then
    TAE.Tae_Wpt.Wpt_NewPanel
        ( Dummy
          Data_Vm
          View_Vm
          Relative_Window
          User_Context
          Flags
          Panel_Id
          => "",
          => Info.Target,
          => Info.View,
          => Relative_Window,
          => Info,
          => Panel_State,
          => Info.Panel_Id );
else
    Text_IO.Put_Line ("Panel (select) is already displayed.");
end if;

exception

when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_select.Create_Panel: "
        & "Panel was not initialized prior to creation.");
    raise;

when TAE.TAE_FAIL =>
    Text_IO.Put_Line ("Panel_select.Create_Panel: "
        & "Panel could not be created.");
    raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
    ( Panel_State
      : in TAE.Tae_Wpt.Wpt_Flags
      := TAE.Tae_Wpt.WPT_PREFERRED;

      Relative_Window
      : in X_Windows.Window
      := X_Windows.Null_Window ) is

    Dummy: Boolean;
    L     : Integer:=1;
    label: String(1..80):=(others=>' ');

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then

```

```

        Create_Panel
        ( Relative_Window      => Relative_Window,
          Panel_State          => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;

    --ADDED
    strlen(component,L);
    label(1..L):=component(1..L);

TAE.Tae_Vm.Vm_SetString(Info.View,"label",1,label(1..L),TAE.Tae_Vm.P_UPD
ATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"label",Info.View,"label");
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_select.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .    Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_select.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

```

```

end Destroy_Panel;

-----
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .   delete_Event           -- Subprogram SPEC & BODY
-- .
-- .....

procedure delete_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- delete_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  --
  -- End default generated code

  -- Begin generated code for Connection
  --
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
  Panel_cdelwarn.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
  --
  -- End generated code for Connection

end delete_Event;

-- .....
-- .
-- .   save_Event             -- Subprogram SPEC & BODY
-- .
-- .....

procedure save_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

```



```

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;
L,I,C : Integer:=1;
D      : Integer:=0;
ls_file :file_type;
file_name:String(1..7) := "ls_file";
current :String (1..80):= (others=>' ');
number  :integer:=1;
Dummy   :Boolean;

begin -- save_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    --ADDED
    strlen(path,D);
    Directory(1..D):=path(1..D);
    loop
        if Directory(I) = '/' then
            current_directory(cur_dir_index)(1..C+1):=current(1..C-
1)&"/ ";
            cur_dir_index:=cur_dir_index+1;
            C:=1;
        else
            current(C):=Directory(I);
            C:=C+1;
        end if;
        I:=I+1;
        if I=D+1 then exit;
        end if;
    end loop;
    current_directory(cur_dir_index)(1..C+1):=current(1..C-1)&"/ ";
    if D>27 then
        directory_array := Directory((D-26)..D);
    else
        directory_array := Directory(1..27);
    end if;
    system_call("ls -F "&Directory(1..D)&" > "&"ls_file");
    list_directory(ls_file,file_name,file_vec,number);

```

```

        strlen(component,L);

TAE.Tae_Vm.Vm_SetString(Panel_savecomp.Info.View,"compname",1,component(
1..L),TAE.Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Panel_savecomp.Info.Panel_Id,"compname",Panel
_savecomp.Info.View,"compname");

TAE.Tae_Wpt.Wpt_SetStringConstraints(Panel_savecomp.Info.Panel_Id,"direc
tory",taeint(number),file_vec);

TAE.Tae_Vm.Vm_SetString(Panel_savecomp.Info.View,"directorylabel",1,dire
ctory_array,TAE.Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Panel_savecomp.Info.Panel_Id,"directorylabel"
,Panel_savecomp.Info.View,"directorylabel");
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    system_call("rm ls_file");
    system_call("rm directory_file");

    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_savecomp.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    --
    -- End generated code for Connection

end save_Event;

-- .....
-- .
-- .    print_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure print_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;
    --ADDED
    J,P:Integer:=1;

begin -- print_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then

```

```

    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
end if;
--
-- End default generated code
    strlen(component,P);
    strlen(library,J);
    system_call(com&"cv "&library(1..J)&" "&component(1..P)&" outpsdl
outspec outbody");

    if TAE.Tae_Misc.s_equal (Value(1),"PSDL") then
        system_call("lpr outpsdl");
        TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"SPOOLED TO DEFAULT
PRINTER");
    elsif TAE.Tae_Misc.s_equal (Value(1), "Spec") then
        system_call("lpr outspec");
        TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"SPOOLED TO DEFAULT
PRINTER");
    elsif TAE.Tae_Misc.s_equal (Value(1), "Body") then
        system_call("lpr outbody");
        TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"SPOOLED TO DEFAULT
PRINTER");
    end if;
    system_call("rm outpsdl");
    system_call("rm outspec");
    system_call("rm outbody");

end print_Event;

-- .....
-- .
-- .   integrate_Event           -- Subprogram SPEC & BODY
-- .
-- .....

procedure integrate_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is
    ,
--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;
    L,S,Q : Integer:=1;
    teststring: String(1..7):= (others=>' ');
    genfile   : File_type;

begin -- integrate_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);

```

```

if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
end if;
--ADDED

    strlen(component,L);
    strlen(library,S);
    strlen(Query_psd1,Q);
    system_call(com&"cv "&library(1..S)&" "&component(1..L)&"
retrieved_psd1 retrieved_spec retrieved_body");
    if Query then
        system_call(parse&"main retrieved_psd1 "&Query_psd1(1..Q));
    else
        system_call(parse&"parse retrieved_psd1");
    end if;

    open(genfile,mode=>in_file,name=>"genfile");
    get(genfile,teststring);
    close(genfile);
    if teststring(1..7) = "generic" then
        system_call("cp softbase_txt.gen message");
    else
        system_call("cp softbase_txt.nongen message");
    end if;

    for erase in 1..7 loop
        teststring(erase):= ' ';
    end loop;
    system_call("rm genfile");
    --
    -- End default generated code

    -- Begin generated code for Connection
    --
    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_integrt.Connect_Panel (TAE.Tae_Wpt.WPT_PREFERRED);
    --
    -- End generated code for Connection

end integrate_Event;

-- .....
-- .
-- .    cancel_Event                -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:

```

```

--| EVENT HANDLER.  Insert application specific information.

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;

begin -- cancel_Event
  -- Begin default generated code
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  --
  -- End default generated code
  -- Begin generated code for Connection
  --
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
  Panel_compsel.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
  -- End generated code for Connection
end cancel_Event;

-- end EVENT HANDLERS

-- .....
-- .
-- .    Dispatch_Item                -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
  ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

  if TAE.Tae_Misc.s_equal ("delete", User_Context_Ptr.Parm_Name) then
    delete_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("save", User_Context_Ptr.Parm_Name) then
    save_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("print", User_Context_Ptr.Parm_Name) then
    print_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("integrate", User_Context_Ptr.Parm_Name)
then
    integrate_Event (User_Context_Ptr);
  elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
    cancel_Event (User_Context_Ptr);
  end if;

  end Dispatch_Item;

end Panel_select;

```

Q. PANEL_SAVECOMP PACKAGE

1. PAN_SAVECOMP_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_savecomp.s.a
-- *** Generated  : Aug 12 11:50:32 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_savecomp                      -- Package SPEC
-- *
-- *****
```

```
with TAE;
with X_Windows;
```

package Panel_savecomp is

```
--| PURPOSE:
--| This package encapsulates the TAE Plus panel: savecomp
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     savecomp
```

```
Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information
```

```
-- .....
-- .
-- .      Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....
```

```
procedure Initialize_Panel
( Collection_Read          -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr ); -- resource file
```

```
--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
  ( Panel_State                               -- Flags sent to Wpt_NewPanel.
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window                          -- Panel origin is offset from
      : in X_Windows.Window                  -- this X Window.  Null_Window
      := X_Windows.Null_Window );           -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                               -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window                          -- Panel origin is offset from
      : in X_Windows.Window                  -- this X Window.  Null_Window
      := X_Windows.Null_Window );           -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specifiec Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--| not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--| created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--| Panel_State is an invalid state

-- .....
-- .
-- . Destroy_Panel -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- . Dispatch_Item -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr -- Wpt Event Context for a PARM
: in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_savecomp;

```


2. PAN_SAVECOMP_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File:      pan_savecomp_b.a
-- *** Generated:  Aug 12 11:50:32 1992
-- *****
-- *
-- *      Panel_savecomp                      -- Package BODY
-- *
-- *****

with TAE; use TAE;
with Text_IO;
with Global;
use Global,Text_IO;

-- One "with" statement for each connected panel.
with Panel_select;

package body Panel_savecomp is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|      The panel's name is changed (not title)
--| For panel:
--|      savecomp
--|
--| The following WorkBench operations will also cause regeneration:
--|      An item is deleted
--|      A new item is added to this panel
--|      An item's name is changed (not title)
--|      An item's data type is changed
--|      An item's generates events flag is changed
--|      An item's valids changed (if item is type string and connected)
--|      An item's connection information changed
--| For the panel items:
--|      directorylabel,  cancel,          ok,          compname,
--|      directory
```

```

--ADDED
selected :String (1..80):= (others=>' ');
ls_file  :file_type;
file_name:String(1..7) := "ls_file";
number   :integer:=1;

-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
  ( Collection_Read
    : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "savecomp_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "savecomp_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_savecomp.Initialize_Panel:  "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_savecomp.Initialize_Panel:  "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel                          -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window

```

```

        : in X_Windows.Window
        := X_Windows.Null_Window ) is

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
        ( Dummy              => "",
          Data_Vm             => Info.Target,
          View_Vm             => Info.View,
          Relative_Window     => Relative_Window,
          User_Context        => Info,
          Flags               => Panel_State,
          Panel_Id            => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (savecomp) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_savecomp.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_savecomp.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .   Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

begin -- Connect_Panel

```

```

if Info.Panel_Id = Tae.Null_Panel_Id then
  Create_Panel
    ( Relative_Window      => Relative_Window,
      Panel_State          => Panel_State );
else
  TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
end if;

exception

when TAE.Tae_Wpt.BAD_STATE =>
  Text_IO.Put_Line ("Panel_savecomp.Connect_Panel:  "
    & "Invalid panel state.");
  raise;

end Connect_Panel;

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

--| NOTES: (none)

begin -- Destroy_Panel

  TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

when TAE.Tae_Wpt.BAD_PANEL_ID =>
  Text_IO.Put_Line ("Panel_savecomp.Destroy_Panel:  "
    & "Info.Panel_Id is an invalid id.");
  raise;

when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
  -- This panel has not been created yet, or has already been destroyed.
  -- Trap this exception and do nothing.
  null;

end Destroy_Panel;

--+++++
--
-- begin EVENT HANDLERS
--

```

```

-- .....
-- .
-- .      cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;

begin -- cancel_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  --
  -- End default generated code

  -- Begin generated code for Connection
  --
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
  Panel_select.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
  --
  -- End generated code for Connection
  --ADDED
  TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);
  for J in 1..80 loop
    Directory(J):=' ';
  end loop;

  for K in 1..cur_dir_index loop
    for L in 1..80 loop
      current_directory(K)(L):=' ';
    end loop;
  end loop;
  for A in 1..27 loop
    directory_array(A):=' ';
  end loop;
  cur_dir_index:=1;

end cancel_Event;

```

```

-- .....
-- .
-- .      ok_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure ok_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.
  --|
  --| NOTES: (none)

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;
  --ADDED
  N,K,J,P:Integer:=1;

begin -- ok_Event

  -- Begin default generated code
  --
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  --ADDED
  strlen(selected,N);
  strlen(Directory,K);
  strlen(library,J);
  strlen(component,P);

  system_call(com&"cv "&library(1..J)&" "&component(1..P)&" outpsdl
outspec outbody");
  system_call("cp outpsdl "&Directory(1..K)&"/
"&component(1..P)&".psdl");
  system_call("cp outspec "&Directory(1..K)&"/
"&component(1..P)&".spec.a");
  system_call("cp outbody "&Directory(1..K)&"/
"&component(1..P)&".body.a");
  system_call("rm outpsdl");
  system_call("rm outspec");
  system_call("rm outbody");

  --
  -- End default generated code

  -- Begin generated code for Connection
  --
  Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);

```

```

Panel_select.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection
--ADDED
TAE.Tae_Wpt.Wpt_PanelReset(Info.Panel_Id);
for X in 1..80 loop
    Directory(X):=' ';
end loop;

for Y in 1..cur_dir_index loop
    for Z in 1..80 loop
        current_directory(Y)(Z):=' ';
    end loop;
end loop;
for AA in 1..27 loop
    directory_array(AA):=' ';
end loop;
cur_dir_index:=1;

end ok_Event;

-- .....
-- .
-- .    directory_Event                -- Subprogram SPEC & BODY
-- .
-- .....

procedure directory_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.
--|
--| NOTES: (none)

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;
--ADDED
N,M,K,S : integer:=1;
Dummy:Boolean;
selected_vec:s_vector(1..1):= (others=> new STRING(1..80));

begin -- directory_Event

-- Begin default generated code
--
TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
end if;
--ADDED

```

```

selected:=Value(1)(1..80);
parse_line(selected);
strlen(selected,N);
strlen(Directory,M);
if Is_a_directory then
    K:=N+M;
    Directory(1..K):=Directory(1..M)&"/"&selected(1..(N-1));

    if K>27 then
        directory_array := Directory((K-26)..K);
    else
        directory_array := Directory(1..27);
    end if;

TAE.Tae_Vm.Vm_SetString(Info.View,"directorylabel",1,directory_array,TAE
.Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"directorylabel",Info.View,"dir
ectorylabel");

    system_call("ls -F "&Directory(1..K)&" > "&ls_file");
    list_directory(ls_file,file_name,file_vec,number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"directory",taeint(nu
mber),file_vec);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    cur_dir_index:=cur_dir_index+1;
    Current_directory(cur_dir_index)(1..N):=selected(1..N);
    for node in (N+1)..80 loop
        Current_directory(cur_dir_index)(node):=' ';
    end loop;
    system_call("rm ls_file");
    Is_a_directory:=FALSE;

elseif Upper_directory then
    strlen(Current_directory(cur_dir_index),S);
    for drct in (M-S+1)..80 loop
        Directory(drct):=' ';
    end loop;

    if (M-S)>27 then
        directory_array := Directory(((M-S)-26)..(M-S));
    else
        directory_array := Directory(1..27);
    end if;

TAE.Tae_Vm.Vm_SetString(Info.View,"directorylabel",1,directory_array,TAE
.Tae_Vm.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"directorylabel",Info.View,"dir
ectorylabel");

```



```

        cur_dir_index:=cur_dir_index-1;
        system_call("ls -F "&Directory(1..(M-S))&" > "&"ls_file");
        list_directory(ls_file,file_name,file_vec,number);

TAE.Tae_Wpt.Wpt_SetStringConstraints(Info.Panel_Id,"directory",taeint(nu
mber),file_vec);
        Dummy:=TAE.Tae_Wpt.Wpt_Pending;
        system_call("rm ls_file");
        Upper_directory:=FALSE;

        else --A file name is selected
            TAE.Tae_Wpt.Wpt_PanelMessage(info.panel_id,"SELECTION IS NOT A
DIRECTORY");
            end if;

--
-- End default generated code

end directory_Event;

--
-- end EVENT HANDLERS
--
-----
-- .....
-- .
-- .      Dispatch_Item                      -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

begin -- Dispatch_Item

    if TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("ok", User_Context_Ptr.Parm_Name) then
        ok_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("directory", User_Context_Ptr.Parm_Name)
then
        directory_Event (User_Context_Ptr);
        end if;

    end Dispatch_Item;

end Panel_savecomp;

```

R. PANEL_CDELWARN PACKAGE

1. PAN_CDELWARN_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_cdelwarn_s.a
-- *** Generated  : Aug 12 11:50:32 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_cdelwarn                      -- Package SPEC
-- *
-- *****

with TAE;
with X_Windows;

package Panel_cdelwarn is

--| PURPOSE:
--| This package encapsulates the TAE Plus panel: cdelwarn
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching. For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   cdelwarn

Info : TAE.Tae_Wpt.Event_Context_Ptr;  -- panel information

-- .....
-- .
-- .   Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Initialize_Panel
( Collection_Read                      -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr );  -- resource file

--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
--|
```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read

-- .....
-- .
-- .   Create_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Create_Panel
( Panel_State                      -- Flags sent to Wpt_NewPanel.
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created

-- .....
-- .
-- .   Connect_Panel                     -- Subprogram SPEC
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window                  -- Panel origin is offset from
  : in X_Windows.Window            -- this X Window.  Null_Window
  := X_Windows.Null_Window );      -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specified Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--|   In this case, Relative_Window is ignored.
--|
--| EXCEPTIONS:

```

```

--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--| not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--| created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--| Panel_State is an invalid state

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram SPEC
-- .
-- .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

-- .....
-- .
-- .   Dispatch_Item                      -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Item
( User_Context_Ptr                      -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_cdelwarn;

```

2. PAN_CDELWARN_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_cdelwarn_b.a
-- *** Generated  : Aug 12 11:50:32 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_cdelwarn                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Global, Text_IO;

-- One "with" statement for each connected panel.
with Panel_mainmenu;
with Panel_select;

package body Panel_cdelwarn is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below. Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler. (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   cdelwarn
--|
--| The following WorkBench operations will also cause regeneration:
--|   An item is deleted
--|   A new item is added to this panel
--|   An item's name is changed (not title)
--|   An item's data type is changed
--|   An item's generates events flag is changed
--|   An item's valids changed (if item is type string and connected)
--|   An item's connection information changed
--| For the panel items:
--|   ok,                cancel
```

```

-- .....
-- .
-- .   Initialize_Panel           -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
( Collection_Read
  : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "cdelwarn_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "cdelwarn_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_cdelwarn.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_cdelwarn.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .   Create_Panel             -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

```

```

begin -- Create_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        TAE.Tae_Wpt.Wpt_NewPanel
            ( Dummy                => "",
              Data_Vm              => Info.Target,
              View_Vm              => Info.View,
              Relative_Window      => Relative_Window,
              User_Context         => Info,
              Flags                => Panel_State,
              Panel_Id             => Info.Panel_Id );
    else
        Text_IO.Put_Line ("Panel (cdelwarn) is already displayed.");
    end if;

exception

    when TAE.UNINITIALIZED_PTR =>
        Text_IO.Put_Line ("Panel_cdelwarn.Create_Panel: "
            & "Panel was not initialized prior to creation.");
        raise;

    when TAE.TAE_FAIL =>
        Text_IO.Put_Line ("Panel_cdelwarn.Create_Panel: "
            & "Panel could not be created.");
        raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
    ( Panel_State
      : in TAE.Tae_Wpt.Wpt_Flags
      := TAE.Tae_Wpt.WPT_PREFERRED;

      Relative_Window
      : in X_Windows.Window
      := X_Windows.Null_Window ) is

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel
            ( Relative_Window      => Relative_Window,
              Panel_State         => Panel_State );
    end if;

```

```

else
    TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
end if;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_cdelwarn.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .   Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

exception

    when TAE.Tae_Wpt.BAD_PANEL_ID =>
        Text_IO.Put_Line ("Panel_cdelwarn.Destroy_Panel:  "
            & "Info.Panel_Id is an invalid id.");
        raise;

    when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
        -- This panel has not been created yet, or has already been destroyed.
        -- Trap this exception and do nothing.
        null;

end Destroy_Panel;

-----
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .   ok_Event                          -- Subprogram SPEC & BODY
-- .
-- .....

procedure ok_Event

```



```

( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;
L,S   : Integer:=1;

begin -- ok_Event

-- Begin default generated code
--
TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
else
    null;
end if;
--ADDED
strlen(component,L);
strlen(library,S);
system_call(com&"cd "&library(1..S)&" "&component(1..L));
TAE.Tae_Wpt.Wpt_PanelReset (Panel_mainmenu.Info.Panel_Id);
for Clean in 1..(num_of_comp) loop
    for ch in 1..80 loop
        file_vec(Clean).all(ch):=' ';
    end loop;
end loop;

--
-- End default generated code

-- Begin generated code for Connection
--
Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
Panel_mainmenu.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
--
-- End generated code for Connection

end ok_Event;

-- .....
-- .
-- .   cancel_Event                               -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

```

```

--| PURPOSE:
--| EVENT HANDLER.  Insert application specific information.

Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
Count : TAE.Taeint;

begin -- cancel_Event

    -- Begin default generated code
    --
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    else
        null;
    end if;
    --
    -- End default generated code
    -- Begin generated code for Connection
    Connect_Panel (TAE.Tae_Wpt.WPT_INVISIBLE);
    Panel_select.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    --
    -- End generated code for Connection

end cancel_Event;
-- end EVENT HANDLERS
--

-- .....
-- .
-- .    Dispatch_Item                -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
    ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is
    ,
--| NOTES: (none)

begin -- Dispatch_Item

    if TAE.Tae_Misc.s_equal ("ok", User_Context_Ptr.Parm_Name) then
        ok_Event (User_Context_Ptr);
    elsif TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
        cancel_Event (User_Context_Ptr);
    end if;

end Dispatch_Item;

end Panel_cdelwarn;

```

S. PANEL_INTEGRATE PACKAGE

1. PAN_INTEGRT_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_integrt_s.a
-- *** Generated  : Sep 13 13:51:07 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      Panel_integrt                      -- Package SPEC
-- *
-- *****
```

with TAE;
with X_Windows;

package Panel_integrt is

```
--| PURPOSE:
--| This package encapsulates the TAE Plus panel:  integrt
--| These subprograms enable panel initialization, creation, destruction,
--| and event dispatching.  For more advanced manipulation of the panel
--| using the TAE package, the panel's Event_Context (Info) is provided.
--| It includes the Target and View (available after initialization)
--| and the Panel_Id (available after creation).
--|
--| REGENERATED:
--| The following Workbench operations will cause regeneration of this
--| file:
--|     The panel's name is changed (not title)
--| For panel:
--|     integrt
```

Info : TAE.Tae_Wpt.Event_Context_Ptr; -- panel information

```
-- .....
-- .
-- .      Initialize_Panel                      -- Subprogram SPEC
-- .
-- .....
```

```
procedure Initialize_Panel
( Collection_Read                      -- TAE Collection read from
  : in TAE.Tae_Co.Collection_Ptr );  -- resource file
```

```
--| PURPOSE:
--| This procedure initializes the Info.Target and Info.View for this
--| panel
```

```

--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if Collection_Read not initialized
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised if the panel is not in
--|   Collection_Read
--|
--| .....
--| .
--|   Create_Panel                                -- Subprogram SPEC
--| .
--| .....

procedure Create_Panel
  ( Panel_State                                -- Flags sent to Wpt_NewPanel.
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window                          -- Panel origin is offset from
    : in X_Windows.Window                    -- this X Window. Null_Window
    := X_Windows.Null_Window );              -- uses the root window.

--| PURPOSE:
--| This procedure creates this panel object in the specified Panel_State
--| and stores the panel Id in Info.Panel_Id.
--|
--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised if the panel is not initialized
--| TAE.TAE_FAIL is raised if the panel could not be created
--|
--| .....
--| .
--|   Connect_Panel                                -- Subprogram SPEC
--| .
--| .....

procedure Connect_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window                          -- Panel origin is offset from
    : in X_Windows.Window                    -- this X Window. Null_Window
    := X_Windows.Null_Window );              -- uses the root window.

--| PURPOSE:
--| If this panel doesn't exist, this procedure creates this panel object
--|   in the specifiec Panel_State and stores the panel Id in
--|   Info.Panel_Id.
--| If this panel does exist, it is set to the specified Panel_State.
--| In this case, Relative_Window is ignored.
--|

```

```

--| EXCEPTIONS:
--| TAE.UNINITIALIZED_PTR is raised from Create_Panel if the panel is
--|   not initialized
--| TAE.TAE_FAIL is raised from Create_Panel if the panel could not be
--|   created
--| TAE.Tae_Wpt.BAD_STATE is raised if the panel exists and the
--|   Panel_State is an invalid state

--| .....
--| .
--|   Destroy_Panel                      -- Subprogram SPEC
--| .
--| .....

procedure Destroy_Panel;

--| PURPOSE:
--| This procedure erases a panel from the screen and de-allocates the
--| associated panel object (not the target and view).
--|
--| EXCEPTIONS:
--| TAE.Tae_Wpt.BAD_PANEL_ID is raised if Info.Panel_Id is an invalid
id.
--|
--| NOTES:
--| Info.Panel_Id is set to TAE.NULL_PANEL_ID, and should not referenced
--| in any Wpt call until it is created again.

--| .....
--| .
--|   Dispatch_Item                      -- Subprogram SPEC
--| .
--| .....

procedure Dispatch_Item
  ( User_Context_Ptr                      -- Wpt Event Context for a PARM
    : in TAE.Tae_Wpt.Event_Context_Ptr ); -- event.

--| PURPOSE:
--| This procedure calls the Event Handler specified by User_Context_Ptr
--|
--| EXCEPTIONS:
--| Application-specific

end Panel_integrt;

```

2. PAN_INTEGRT_B.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : pan_integrt_b.a
-- *** Generated  : Oct 22 16:29:54 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *   Panel_integrt                      -- Package BODY
-- *
-- *****
with TAE; use TAE;
with Text_IO;
with Global;
use Text_IO,Global;

-- One "with" statement for each connected panel.
with Panel_select;
with Panel_mainmenu;

package body Panel_integrt is

--| NOTES:
--| For each parameter that you have defined to be "event-generating" in
--| this panel, there is an event handler procedure below.  Each handler
--| has a name that is a concatenation of the parameter name and "_Event".
--| Add application-dependent logic to each event handler.  (As generated
--| by the WorkBench, each event handler simply logs the occurrence of
--| the event.)
--|
--| You may want to flag any changes you make to this file so that if
--| you regenerate this file, you can more easily cut and paste your
--| modifications back in.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|   The panel's name is changed (not title)
--| For panel:
--|   integrt
--|
--| The following WorkBench operations will also cause regeneration:
--|   An item is deleted
--|   A new item is added to this panel
--|   An item's name is changed (not title)
--|   An item's data type is changed
--|   An item's generates events flag is changed
--|   An item's valids changed (if item is type string and connected)
--|   An item's connection information changed
--| For the panel items:
--|   cancel,          integrate,          int_quit
```

```

-- .....
-- .
-- .      Initialize_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Initialize_Panel
  ( Collection_Read
    : in TAE.Tae_Co.Collection_Ptr ) is

begin -- Initialize_Panel

  Info := new TAE.Tae_Wpt.Event_Context;
  Info.Collection := Collection_Read;
  TAE.Tae_Co.Co_Find (Info.Collection, "integrt_v", Info.View);
  TAE.Tae_Co.Co_Find (Info.Collection, "integrt_t", Info.Target);

exception

  when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_integrt.Initialize_Panel: "
      & "Collection_Read not initialized.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    Text_IO.Put_Line ("Panel_integrt.Initialize_Panel: "
      & "(View or Target) not in Collection.");
    raise;

end Initialize_Panel;

-- .....
-- .
-- .      Create_Panel                          -- Subprogram BODY
-- .
-- .....

procedure Create_Panel
  ( Panel_State
    : in TAE.Tae_Wpt.Wpt_Flags
    := TAE.Tae_Wpt.WPT_PREFERRED;

    Relative_Window
    : in X_Windows.Window
    := X_Windows.Null_Window ) is

begin -- Create_Panel

```

```

if Info.Panel_Id = Tae.Null_Panel_Id then
    TAE.Tae_Wpt.Wpt_NewPanel
        ( Dummy          => "",
          Data_Vm         => Info.Target,
          View_Vm         => Info.View,
          Relative_Window => Relative_Window,
          User_Context    => Info,
          Flags           => Panel_State,
          Panel_Id        => Info.Panel_Id );
else
    Text_IO.Put_Line ("Panel (integrt) is already displayed.");
end if;

exception

when TAE.UNINITIALIZED_PTR =>
    Text_IO.Put_Line ("Panel_integrt.Create_Panel: "
        & "Panel was not initialized prior to creation.");
    raise;

when TAE.TAE_FAIL =>
    Text_IO.Put_Line ("Panel_integrt.Create_Panel: "
        & "Panel could not be created.");
    raise;

end Create_Panel;

-- .....
-- .
-- .    Connect_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Connect_Panel
( Panel_State
  : in TAE.Tae_Wpt.Wpt_Flags
  := TAE.Tae_Wpt.WPT_PREFERRED;

  Relative_Window
  : in X_Windows.Window
  := X_Windows.Null_Window ) is

filename : String(1..80):=(others=>' ');
Q,L,F,P,K: Integer:=1;
Dummy    : Boolean;

begin -- Connect_Panel

    if Info.Panel_Id = Tae.Null_Panel_Id then
        Create_Panel

```



```

        ( Relative_Window      => Relative_Window,
          Panel_State           => Panel_State );
    else
        TAE.Tae_Wpt.Wpt_SetPanelState (Info.Panel_Id, Panel_State);
    end if;
    --ADDED
    strlen(Query_psd1,Q);
    strlen(component,L);
    strlen(proto_prefix,P);
    K:=P+1;
    filename(1..K):=proto_prefix(1..P)&". ";
    if Query then
        --filename(K..(2+Q)):= Query_psd1(1..(Q-4));
        filename(K+1..(K+(Q-3))):= Query_psd1(1..(Q-4))&"a";
        --filename(3+Q):='a';
    else
        filename(K+1..(L+K+2)):= component(1..L)&".a";
    end if;
    strlen(filename,F);

TAE.Tae_Vm.Vm_SetString(Info.View,"filename",1,filename(1..F),TAE.Tae_Vm
.P_UPDATE);

TAE.Tae_Wpt.Wpt_ViewUpdate(Info.Panel_Id,"filename",Info.View,"filename"
);
    Dummy:=TAE.Tae_Wpt.Wpt_Pending;
    for clean in 1..80 loop
        filename(clean):=' ';
    end loop;

exception

    when TAE.Tae_Wpt.BAD_STATE =>
        Text_IO.Put_Line ("Panel_integrt.Connect_Panel:  "
            & "Invalid panel state.");
        raise;

end Connect_Panel;

-- .....
-- .
-- .      Destroy_Panel                      -- Subprogram BODY
-- .
-- .....

procedure Destroy_Panel is

begin -- Destroy_Panel

    TAE.Tae_Wpt.Wpt_PanelErase(Info.Panel_Id);

```

```

exception
  when TAE.Tae_Wpt.BAD_PANEL_ID =>
    Text_IO.Put_Line ("Panel_integrt.Destroy_Panel: "
      & "Info.Panel_Id is an invalid id.");
    raise;

  when TAE.Tae_Wpt.ERASE_NULL_PANEL =>
    -- This panel has not been created yet, or has already been destroyed.
    -- Trap this exception and do nothing.
    null;

end Destroy_Panel;

-----
--
-- begin EVENT HANDLERS
--
-- .....
-- .
-- .   cancel_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure cancel_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;
  Q,L   : Integer:=1;

begin -- cancel_Event
  -- Begin default generated code
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  -- End default generated code
  --ADDED
  system_call("rm retrieved_psd1");
  system_call("rm retrieved_spec");
  system_call("rm retrieved_body");
  system_call("rm outfile");
  system_call("rm message");
  -- Begin generated code for Connection
  Destroy_Panel;
  Panel_select.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
  -- End generated code for Connection
end cancel_Event;

```

```

-- .....
-- .
-- .   integrate_Event   -- Subprogram SPEC & BODY
-- .
-- .....

procedure integrate_Event
  ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

  --| PURPOSE:
  --| EVENT HANDLER.  Insert application specific information.
  --|
  --| NOTES: Invokes the code for integration

  Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
  Count : TAE.Taeint;
  Q,X,P,C: Integer:=1;

begin -- integrate_Event
  -- Begin default generated code
  TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
  if Count > 0 then
    TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
  end if;
  --ADDED
  strlen(proto_prefix,X);
  strlen(path,P);
  strlen(component,C);
  if Query then
    strlen(Query_psd1,Q);
    if proto_prefix(1..X)=Query_psd1(1..X) then
      system_call("cp outfile "&path(1..P)&"/"&Query_psd1(1..(Q-
4))&"a");
    else
      system_call("cp outfile "&path(1..P)&"/
&proto_prefix(1..X)&". "&Query_psd1(1..(Q-4))&"a");
    end if;
    system_call("cp retrieved_spec "&path(1..P)&"/
&proto_prefix(1..X)&". "&component(1..C)&"_spec.a");
    system_call("cp retrieved_body "&path(1..P)&"/
&proto_prefix(1..X)&". "&component(1..C)&"_body.a");

    else
      system_call("cp outfile "&path(1..P)&"/
&proto_prefix(1..X)&". "&component(1..C)&"_a");
      system_call("cp retrieved_spec "&path(1..P)&"/
&proto_prefix(1..X)&". "&component(1..C)&"_spec.a");
      system_call("cp retrieved_body "&path(1..P)&"/
&proto_prefix(1..X)&". "&component(1..C)&"_body.a");
    end if;
    Query:=False;

```

```

    for erase in 1..80 loop
        Query_psd1(erase):=' ';
    end loop;
    system_call("rm outfile");
    system_call("rm message");
    system_call("rm retrieved_psd1");
    system_call("rm retrieved_spec");
    system_call("rm retrieved_body");

    -- End default generated code
    -- Begin generated code for Connection
    Destroy_Panel;
    Panel_mainmenu.Connect_Panel (TAE.Tae_Wpt.WPT_VISIBLE);
    -- End generated code for Connection

end integrate_Event;

-- .....
-- .
-- .      int_quit_Event                      -- Subprogram SPEC & BODY
-- .
-- .....

procedure int_quit_Event
    ( Info : in TAE.Tae_Wpt.Event_Context_Ptr ) is

    --| PURPOSE:
    --| EVENT HANDLER.  Insert application specific information.
    --|
    --| NOTES: Invokes the code for integration and terminates the program.

    Value : array (1..1) of String (1..TAE.Tae_Taeconf.STRINGSIZE);
    Count : TAE.Taeint;
    Q,X,P,C: Integer:=1;

begin -- int_quit_Event
    -- Begin default generated code
    TAE.Tae_Vm.Vm_Extract_Count (Info.Parm_Ptr, Count);
    if Count > 0 then
        TAE.Tae_Vm.Vm_Extract_SVAL (Info.Parm_Ptr, 1, Value(1));
    end if;
    --ADDED
    strlen(proto_prefix,X);
    strlen(path,P);
    strlen(component,C);
    if Query then
        strlen(Query_psd1,Q);
        system_call("cp outfile "&path(1..P)&"/
"&proto_prefix(1..X)&". "&Query_psd1(1..(Q-4))&"a");
        system_call("cp retrieved_spec "&path(1..P)&"/
"&proto_prefix(1..X)&". "&component(1..C)&"_spec.a");

```

```

        system_call("cp retrieved_body "&path(1..P)&"/
"&proto_prefix(1..X)&". "&component(1..C)&"_body.a");
        --system_call("rm "&Query_psd1(1..Q));
    else
        system_call("cp outfile "&path(1..P)&"/
"&proto_prefix(1..X)&". "&component(1..C)&".a");
        system_call("cp retrieved_spec "&path(1..P)&"/
"&proto_prefix(1..X)&". "&component(1..C)&"_spec.a");
        system_call("cp retrieved_body "&path(1..P)&"/
"&proto_prefix(1..X)&". "&component(1..C)&"_body.a");
        end if;

        system_call("rm outfile");
        system_call("rm message");
        system_call("rm retrieved_psd1");
        system_call("rm retrieved_spec");
        system_call("rm retrieved_body");
        Global.Set_Application_Done;

    end int_quit_Event;

-- end EVENT HANDLERS

-- .....
-- .
-- .    Dispatch_Item                      -- Subprogram BODY
-- .
-- .....

procedure Dispatch_Item
    ( User_Context_Ptr : in TAE.Tae_Wpt.Event_Context_Ptr ) is

    --| NOTES: (none)

    begin -- Dispatch_Item

        if TAE.Tae_Misc.s_equal ("cancel", User_Context_Ptr.Parm_Name) then
            cancel_Event (User_Context_Ptr);
        elsif TAE.Tae_Misc.s_equal ("integrate", User_Context_Ptr.Parm_Name)
        then
            integrate_Event (User_Context_Ptr);
        elsif TAE.Tae_Misc.s_equal ("int_quit", User_Context_Ptr.Parm_Name)
        then
            int_quit_Event (User_Context_Ptr);
        end if;

    end Dispatch_Item;

end Panel_integrt;

```

T. SOFTBASE_SUPPORT PACKAGE

1. SOFTBASE_SUPPORT_S.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : softbase_support_s.a
-- *** Generated  : May 21 16:12:31 1992
-- *** Revised by : Dogan Ozdemir
-- *****
-- *
-- *      softbase_Support                      -- Package SPEC
-- *
-- *****

with TAE;

package softbase_Support is

--| PURPOSE:
--| This package encapsulates the operations specific to a resource file.
--| These subprograms enable initialization of all panels, displaying
--| all initial panels, and event dispatching.
--|
--| REGENERATED:
--| This file is generated only once.

-- .....
-- .
-- .      Initialize_All_Panels                  -- Subprogram SPEC
-- .
-- .....

procedure Initialize_All_Panels
  ( Resource_File                      -- Name of resource file.
    : in String );

--| PURPOSE:
--| This procedure initializes all the panels in the resource file.
--|
--| EXCEPTIONS:
--| TAE.TAE_FAIL is raised when the resource file could not be read
--| TAE.Tae_Co.NO_SUCH_MEMBER is raised when one of the panels was not in
--| the resource file. This could happen if a panel was deleted from
--| the resource file (using the WorkBench) after the code was
--| generated.
--|
--| NOTES:
--| This procedure reads in the resource file and initializes each panel.
--| It only needs to be called once.
```

```

-- .....
-- .
-- .   Create_Initial_Panels           -- Subprogram SPEC
-- .
-- .....

procedure Create_Initial_Panels;

--| PURPOSE:
--| This procedure displays the set of initial panels.

-- .....
-- .
-- .   Dispatch_Panel                 -- Subprogram SPEC
-- .
-- .....

procedure Dispatch_Panel
( User_Context_Ptr           -- Wpt Event Context for a PARM
  : in TAE.Tae_Wpt.Event_Context_Ptr; -- event.

  Panel_In_Resource_File     -- True, if the PARM event
  : out Boolean );           -- occurred in one of the panels
                             -- of this resource file.
                             -- False, otherwise.

--| PURPOSE:
--| This procedure dispatches a WPT_PARM_EVENT to the appropriate
panel's
--| Dispatch_Item routine.
--|
--| EXCEPTIONS:
--| Application-specific

end softbase_Support;

```

2. SOFTBASE_SUPPORT_B.A

```

-- *** TAE Plus Code Generator version V5.1
-- *** File      : softbase_support_b.a
-- *** Generated  : Sep 14 09:04:39 1992
-- *** Revised by : Dogan Ozdemir
--
*****
-- *
-- *   softbase_Support           -- Package BODY
-- *
--
*****

```

```

with TAE;
with Text_IO;

-- one "with" for each panel in the resource file
with Panel_library;
with Panel_lbselect;
with Panel_libadd;
with Panel_lbdelete;
with Panel_ldelwarn;
with Panel_mainmenu;
with Panel_addfile;
with Panel_compsel;
with Panel_keyword;
with Panel_query;
with Panel_viewpsdl;
with Panel_viewspec;
with Panel_viewbody;
with Panel_select;
with Panel_cdelwarn;
with Panel_savecomp;
with Panel_integrt;

package body softbase_Support is

--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|     A panel is deleted
--|     A new panel is added
--|     A panel's name is changed (not title)
--| For the panels:
--| library, lbselect, libadd, lbdelete, ldelwarn, mainmenu, addfile,
--| compsel, keyword, query, viewpsdl, viewspec, viewbody, select,
--| cdelwarn, savecomp, integrt
--|

-- .....
-- .
-- .   Create_Initial_Panels           -- Subprogram STUB
-- .
-- .....

procedure Create_Initial_Panels is separate;

-- .....
-- .
-- .   Initialize_All_Panels          -- Subprogram BODY
-- .
-- .....

procedure Initialize_All_Panels

```



```

( Resource_File
  : in String ) is

Vm_Collection_Read : TAE.Tae_Co.Collection_Ptr;

begin -- Initialize_All_Panels
  -- do one Co_New and Co_ReadFile per resource file
  TAE.Tae_Co.Co_New
    ( Flags => 0,
      Coid => Vm_Collection_Read );
  -- could pass P_Abort if you prefer
  TAE.Tae_Co.Co_ReadFile
    ( Coid => Vm_Collection_Read,
      Spec => Resource_File,
      Mode => TAE.P_CONT );

  Panel_library.Initialize_Panel (Vm_Collection_Read);
  Panel_lbselect.Initialize_Panel (Vm_Collection_Read);
  Panel_libadd.Initialize_Panel (Vm_Collection_Read);
  Panel_lbdelete.Initialize_Panel (Vm_Collection_Read);
  Panel_ldelwarn.Initialize_Panel (Vm_Collection_Read);
  Panel_mainmenu.Initialize_Panel (Vm_Collection_Read);
  Panel_addfile.Initialize_Panel (Vm_Collection_Read);
  Panel_compsel.Initialize_Panel (Vm_Collection_Read);
  Panel_keyword.Initialize_Panel (Vm_Collection_Read);
  Panel_query.Initialize_Panel (Vm_Collection_Read);
  Panel_viewsddl.Initialize_Panel (Vm_Collection_Read);
  Panel_viewspec.Initialize_Panel (Vm_Collection_Read);
  Panel_viewbody.Initialize_Panel (Vm_Collection_Read);
  Panel_select.Initialize_Panel (Vm_Collection_Read);
  Panel_cdelwarn.Initialize_Panel (Vm_Collection_Read);
  Panel_savecomp.Initialize_Panel (Vm_Collection_Read);
  Panel_integrt.Initialize_Panel (Vm_Collection_Read);

exception
  when TAE.TAE_FAIL =>
    Text_IO.Put_Line ("resfile_Support.Initialize_All_Panels: ");
    Text_IO.Put_Line (Resource_File
      & " doesn't exist or is incorrectly formatted.");
    raise;

  when TAE.Tae_Co.NO_SUCH_MEMBER =>
    -- raised from one of the panel's Initialize_Panel
    raise;

end Initialize_All_Panels;

-- .....
-- .
-- .    Dispatch_Panel                      -- Subprogram BODY
-- .
-- .....

```

```

procedure Dispatch_Panel
( User_Context_Ptr
  : in TAE.Tae_Wpt.Event_Context_Ptr;

  Panel_In_Resource_File
  : out Boolean ) is

begin -- Dispatch_Parm_Event

  Panel_In_Resource_File := TRUE;
  if TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_library.Info) then
    Panel_library.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_lbselect.Info) then
    Panel_lbselect.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_libadd.Info) then
    Panel_libadd.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_lbdelete.Info) then
    Panel_lbdelete.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_ldelwarn.Info) then
    Panel_ldelwarn.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_mainmenu.Info) then
    Panel_mainmenu.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_addfile.Info) then
    Panel_addfile.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_compsel.Info) then
    Panel_compsel.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_keyword.Info) then
    Panel_keyword.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_query.Info) then
    Panel_query.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_viewsddl.Info) then
    Panel_viewsddl.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_viewspec.Info) then
    Panel_viewspec.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_viewbody.Info) then
    Panel_viewbody.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_select.Info) then
    Panel_select.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_cdelwarn.Info) then
    Panel_cdelwarn.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_savecomp.Info) then
    Panel_savecomp.Dispatch_Item (User_Context_Ptr);
  elsif TAE.Tae_Wpt."=" (User_Context_Ptr, Panel_integrt.Info) then
    Panel_integrt.Dispatch_Item (User_Context_Ptr);
  else
    Panel_In_Resource_File := FALSE;
  end if;
end Dispatch_Panel;

end softbase_Support;

```

U. SOFTBASE_CREAT_INIT.A

```
-- *** TAE Plus Code Generator version V5.1
-- *** File      : softbase__creat_init.a
-- *** Generated  : Aug 12 16:02:32 1992
-- *** Revised by : Dogan Ozdemir
-- .....
-- .
-- .   Create_InitialPanels          -- Subprogram SUBUNIT
-- .
-- .....

separate (softbase_Support)
procedure Create_Initial_Panels is

--| NOTES:
--| This subprogram is not in the same file as softbase_Support.a
--| for code regeneration purposes. Therefore it is a subunit.
--| Also note, that the parent unit "with"ed in all of the panel packages
--| for the entire resource file, so this unit doesn't need to "with" in
--| any of the panel packages.
--|
--| This procedure should be called after the panels in the initial panel
--| set have been initialized.
--|
--| REGENERATED:
--| The following WorkBench operations will cause regeneration of this
--| file:
--|   A panel is added to the initial panel set
--|   A panel is deleted from the initial panel set
--| For the set of initial panels:
--|   library, lbdelete, lbselect, addfile, query, keyword, compsel,
--|   savecomp
--|

begin -- Create_Initial_Panels
,
  Panel_library.Create_Panel;
  Panel_lbdelete.Create_Panel;
  Panel_lbselect.Create_Panel;
  Panel_addfile.Create_Panel;
  Panel_query.Create_Panel;
  Panel_keyword.Create_Panel;
  Panel_compsel.Create_Panel;
  Panel_savecomp.Create_Panel;

end Create_Initial_Panels;
```

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 221314	2
Dudley Knox Library Code 52 Naval Postgraduate School Monterey, CA 93943	2
Chairman, Code CS Computer Science Department Naval Postgraduate School Monterey, CA 93943	1
Prof. Yuh-jeng Lee, Code CSLE Computer Science Department Naval Postgraduate School Monterey, CA 93943	2
Prof. Luqi, Code CSLq Computer Science Department Naval Postgraduate School Monterey, CA 93943	2
Deniz Kuvvetleri K.ligi Personel Daire Bsk.ligi Bakanliklar, Ankara/TURKEY	1
Golcuk Tersanesi Komutanligi Golcuk, Kocaeli / TURKEY	1
Deniz Harp Okulu Komutanligi 81704 Tuzla, Istanbul / TURKEY	2
Taskizak Tersanesi Komutanligi Kasimpasa, Istanbul / TURKEY	1
Dogan Ozdemir Pasabayir Mah. 7. Sok. Mehmet Usta Ap. No: 15/2 Bandirma, Balikesir / TURKEY	1